

(NASA-CR-134007) MULTISPECTRAL SCANNER
DATA PROCESSING ALGORITHM DOCUMENTATION
Technical Report, 1 Nov. 1971 - 31 Jan.
1973 (Environmental Research Inst. of
Michigan) 78-86 p HC \$6.00

CSCS 05B

NASA CR- 134007
ERIM 31650-149-T

N73-30347

Unclas
G3/13 13181



MULTISPECTRAL SCANNER DATA PROCESSING ALGORITHM DOCUMENTATION

by

M. Gordon and J. Erickson
Infrared and Optics Division



FORMERLY WILLOW RUN LABORATORIES.
THE UNIVERSITY OF MICHIGAN

prepared for

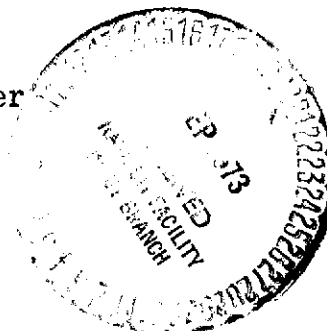
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Lyndon B. Johnson Space Center
NAS 9-9784, Task B 2.14

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
US Department of Commerce
Springfield, VA. 22151

ORIGINAL CONTAINS
COLOR ILLUSTRATIONS

ORIGINAL CONTAINS
COLOR ILLUSTRATIONS



78

NOTICES

Sponsorship. The work reported herein was conducted by the Environmental Research Institute of Michigan* for the National Aeronautics and Space Administration, Lyndon B. Johnson Space Center, under contract NAS 9-9784. Dr. Andrew Potter (TF3) is Technical Monitor. Contracts and grants to the Institute for the support of sponsored research are administered through the Office of Contracts Administration.

Disclaimers. This report was prepared as an account of Government-sponsored work. Neither the United States, nor the National Aeronautics and Space Administration (NASA), nor any person acting on behalf of NASA:

- (A) Makes any warranty or representation, expressed or implied with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
- (B) Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used above, "person acting on behalf of NASA" includes any employee or contractor of NASA, or employee of such contractor, to the extent that such employee or contractor of NASA or employee of such contractor prepares, disseminates, or provides access to any information pursuant to his employment or contract with NASA, or his employment with such contractor.

Availability Notice. Requests for copies of this report should be referred to:

National Aeronautics and Space Administration
Scientific and Technical Information Facility
P. O. Box 33
College Park, Maryland 20740

Final Disposition. After this document has served its purpose, it may be destroyed. Please do not return it to the Environmental Research Institute of Michigan.

* Environmental Research Institute of Michigan
P.O. Box 618
Ann Arbor, Michigan 48107

N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM THE
BEST COPY FURNISHED US BY THE SPONSORING
AGENCY. ALTHOUGH IT IS RECOGNIZED THAT CER-
TAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RE-
LEASED IN THE INTEREST OF MAKING AVAILABLE
AS MUCH INFORMATION AS POSSIBLE.

Technical Report

**MULTISPECTRAL SCANNER DATA PROCESSING
ALGORITHM DOCUMENTATION**

by

M. Gordon and J. Erickson
Infrared and Optics Division



prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

July 1973

NAS 9-9784, Task B 2.14

Lyndon B. Johnson Space Center
Earth Observations Division
Houston, Texas 77058



FOREWORD

This report describes part of a comprehensive and continuing program of research concerned with advancing the state-of-the-art in remote sensing of the environment from aircraft and satellites. The research is being carried out for the NASA's Lyndon B. Johnson Space Center, Houston, Texas, by the Environmental Research Institute of Michigan, formerly the Willow Run Laboratories of The University of Michigan. The basic objective of this multidisciplinary program is to develop remote sensing as a practical tool to provide the planner and decision-maker with extensive information quickly and economically.

Timely information obtained by remote sensing can be important to such people as the farmer, the city planner, the conservationist, and others concerned with problems such as crop yield and disease, urban land studies and development, water pollution, and forest management. The scope of our program includes: (1) extending the understanding of basic processes; (2) discovering new applications, developing advanced remote-sensing systems, and improving automatic data processing to extract information in a useful form; and (3) assisting in data collection, processing, analysis, and ground-truth verification.

The research described here was performed under NASA Contract NAS 9-9784, Task B 2.14 and covers the period from November 1, 1971 through January 31, 1973. Dr. Andrew Potter has been Project Manager. The program was directed by R. R. Legault, Associate Director of the Institute, and by J. D. Erickson, Principal Investigator and Head of the Multispectral Analysis Section. The Institute number for this report is 31650-149-T.



ABSTRACT

The procedures followed in multispectral scanner data evaluation may be divided into two categories: (1) data handling and calibration, and (2) recognition processing and subsequent evaluation of output. The report provides a description for the analyst of the algorithms employed in the current ERIM data processing scheme. Methods for suitable visual display of the results of this processing are also discussed.



CONTENTS

Foreword	iii
Abstract	v
List of Figures	viii
1. Summary	1
2. Introduction	3
3. Analog-Digital Data Conversion	4
3.1. M-7 Scanner and Data Collection	4
3.2. Analog Data and Tape Duplication	4
3.3. Analog-Digital Conversion with Smoothing	10
3.4. AUTOCAL—Evaluation of Channel Skew and Scan Line Slew	14
4. Clamping, Scaling, Deskewing	16
4.1. Clamping	17
4.2. Scaling	17
4.3. Deskewing	18
5. Data Display	19
6. Signature Extraction	23
7. Recognition Processing	28
8. Recognition Mapping	29
Appendix I: Complement Notation	31
Appendix II: Digital Tape Format	32
Appendix III: Program Listings	33
Distribution List	74-75

FIGURES

1. M-7 Scanner Configuration	5-8
2. Scan Line with Enable Gates and Gated Sample Pulses	11
3. Packing of Multispectral Data Output Word	13
4. Example Recognition Display	21
5. Normal Distribution	26
6. Rejection Criteria	26

MULTISPECTRAL SCANNER DATA PROCESSING ALGORITHM DOCUMENTATION

1

SUMMARY

The classification and evaluation procedure of M-7 multispectral scanner data consists, in general, of two stages: pre-classification data handling and reformatting, and digital recognition processing and evaluation.

Pre-classification data handling and reformatting involves the following stage of processing:

- (1) Tape duplication, to make rough corrections for channel skew because of differences between recording and playback tape equipment
- (2) A-D conversion with "smoothing," to convert analog data to digital data of a specified form while combining overlapping data and reducing the instrumental noise level of the data
- (3) Data misalignment correction
 - (a) to evaluate the corrections necessary for channel skew, i.e., the misalignment of analog data on the physical tape because of improper alignment of the recording heads
 - (b) to correct for scan line slew, i.e., the spatial misregistration of resolution elements between scan lines
- (4) Clamping, scaling, and deskewing, to apply the channel deskewing corrections determined by the program AUTOCAL, and subsequently to calibrate dynamically each scan line based on information present in each line

The digital data tape is ready at this point for input into the classification programs. These classification procedures are:

- (1) Visual data display, to allow the user to select training set regions and the actual region to be classified
- (2) Signature extraction, to gather statistical information concerning the distribution of data values for a particular region in the target area in order to establish a decision criterion for the classification process

- (3) Classification, to evaluate a set of resolution elements individually in terms of the signatures previously established, and associate each resolution element evaluated with one of the signatures, along with a measure of the likelihood of correct classification for each element
- (4) Recognition display, to provide a spatially registered visual display of the output of the classification procedure by means of printed characters, with features allowing the reliability of the recognition output data to be evaluated as well as the overall reliability of both the multispectral scanner data and collected statistical signature data used by the classification algorithm

This set of procedures forms a sequence which, in most cases, generates an accurate representation of the distribution of substances and features in the target area of the scan. However, these procedures are not entirely automatic; at each stage the researcher must select various parameters necessary for proper execution of the algorithms defined in this report.

2

INTRODUCTION

Several basic algorithms are implemented in the ERIM multispectral data handling, processing, and evaluation system. This report outlines for the analyst the basic principles of the system without attempting a rigorous justification of the algorithms currently implemented.

Although other types of multispectral scanner data (such as ERTS scanner data) are processed by ERIM, most of the data to be evaluated are obtained from the M-7 scanner [1]. Since data obtained from this instrument are in analog form, a set of manipulations must be performed to put them into a format acceptable to the equipment used in actual processing and evaluation. This process is known as the A-D (analog-digital) conversion process. A-D conversion is not necessary for out-of-house data, since in most instances this information is already in digital form acceptable to our bulk-processing digital computer (a CDC 1604-B).

Section 3 deals with M-7 scanner data collection procedures, subsequent A-D techniques for reformatting the data, and corrections applied in the A-D process to compensate for possible instrumental inconsistencies. Section 4 discusses the methods by which additional corrections, calculated by techniques discussed in Section 3.3, are applied to the data, as well as factors inherent to the data itself. After these data preparation processes are completed, visual displays of the data are generated for use in evaluating the raw data and to aid the user in evaluating the best method available for further data processing. These display methods are discussed in Section 5. Sections 6 and 7 deal with the methods by which the data is evaluated and classified, that is, the methods by which data points are grouped together and associated with a particular class which has been defined by a pre-determined distribution called a signature. Finally, Section 8 deals with the methods by which the results of classification or recognition processing are displayed.

3

ANALOG-DIGITAL DATA CONVERSION**3.1. M-7 SCANNER AND DATA COLLECTION**

The M-7 scanner, housed in a C-47 aircraft along with analog recording equipment for bulk data storage, is a device which senses the radiation from an area and, by means of a system of optics, divides this radiation into a number of spectral regions (channels) covering the visible through infrared regions of the spectrum. Figure 1 outlines the general configuration of the M-7 scanner system and peripheral equipment.

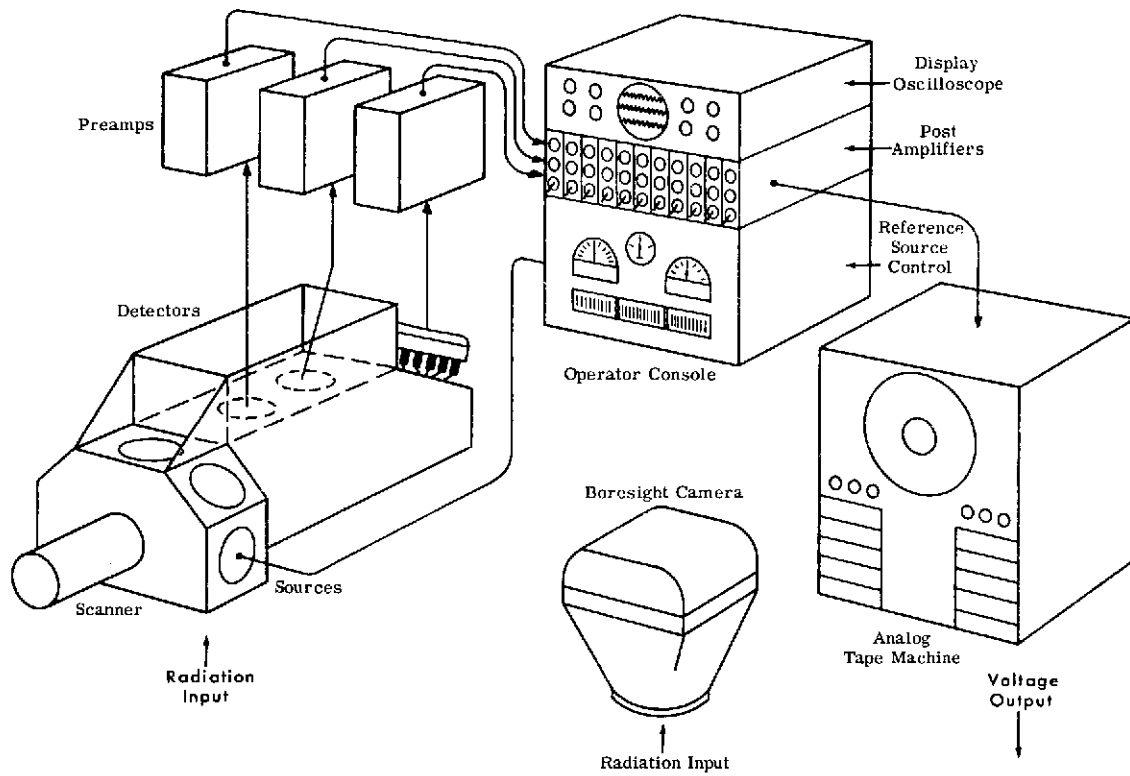
A scan mirror, internal to the scanner housing, is rotated in a plane essentially perpendicular to the direction of flight (see Fig. 1c), such that the sensors within the scanner are sequentially viewing (1) the scene below, (2) one of several internal calibration sources, or (3) the internal scanner housing, which provides a measure of the dark level reading of the sensors. Each rotation of the scan mirror generates a set of data called a scan line and effectively contains a measure of the radiation emanating from a swath of target area, such that the largest dimension of this rectangular swath is perpendicular to the direction of flight. This information from each of the spectral channels, along with additional synchronization information, is recorded continuously on the analog tape. After a desired set of successive scans (referred to as a run) has been recorded on an analog tape, the tape is brought to the ERIM data processing facility for digitization and subsequent processing.

3.2. ANALOG DATA AND TAPE DUPLICATION

The M-7 multispectral scanner data is recorded on analog tape and must go through A-D conversion to produce a digital equivalent compatible with the CDC 1604-B computer.

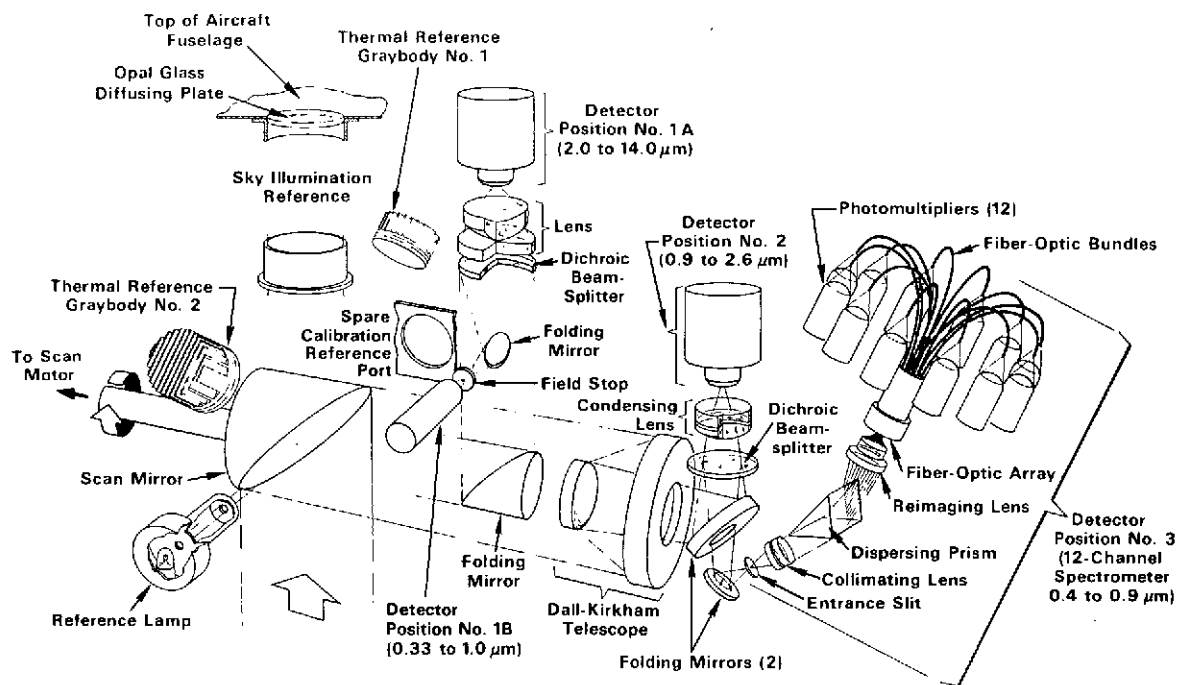
Because of the large quantity of data recorded on the analog tape, only certain areas of the total scan line are usually digitized. A CRT x-y display is used for a C-scan presentation of one channel at a time in the video region of the analog tape (the region where the sensor was actually viewing the scene and not some internal portion of the scanner). This C-scan presentation essentially provides a visual display of the radiation recorded by the scanner, making gross target features apparent. As this C-scan is being presented, a line count is also displayed so that scan line numbers for certain regions of interest can be identified for use in the digitizing procedure.

As the scan mirror rotates, a separate channel of information containing two synchronization pulses per scan line is recorded on the analog tape (Fig. 1d). One of these pulses, the



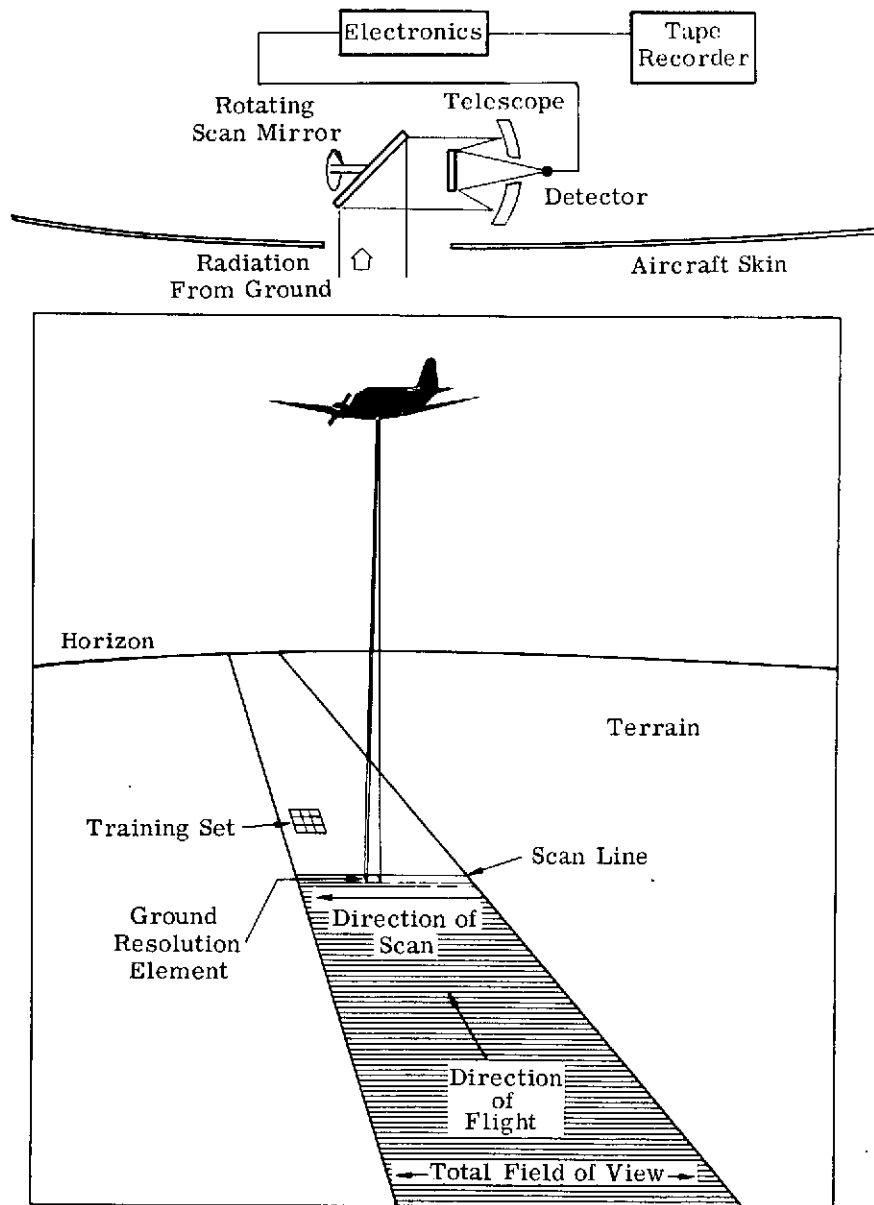
(a) Scanner System General Configuration

FIGURE 1. M-7 SCANNER CONFIGURATION



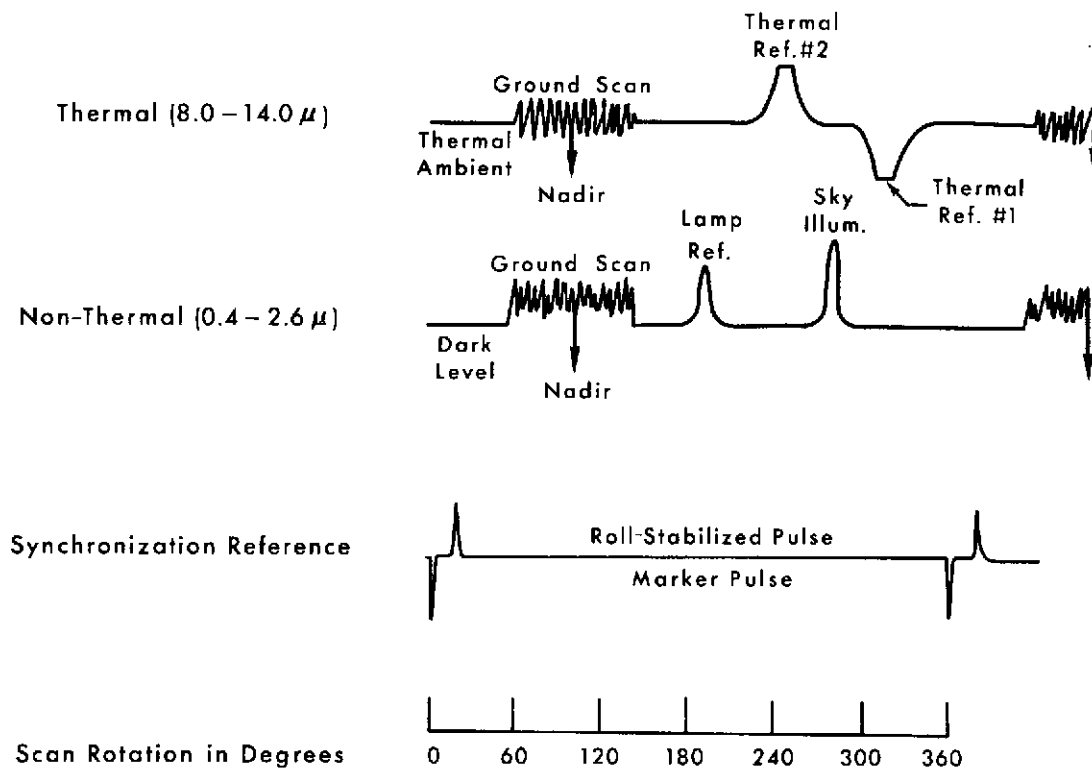
(b) Schematic Diagram of M-7 Scanner

FIGURE 1. M-7 SCANNER CONFIGURATION (Continued)



(c) Geometry of Airborne Scanning

FIGURE 1. M-7 SCANNER CONFIGURATION (Continued)



(d) Scanner Output Voltage Versus Time

FIGURE 1. M-7 SCANNER CONFIGURATION (Concluded)

marker sync pulse, denotes the time when the rotating scan mirror passes a fixed point relative to the scanner housing. The other sync pulse, called the roll stabilized pulse, indicates the time when the scan mirror passes a fixed point relative to the nadir. From the difference in the occurrence time of these two pulses, it is possible to measure the amount of aircraft roll from scan line to scan line. These differences may be digitized along with the multispectral data with special equipment that generates a separate channel called the time channel.

The tape recorder heads used to record the analog data in the aircraft generally have a different orientation relative to the direction of tape movement than the tape recorder used to play back the analog tape to the A-D converter. Because of this, a duplicate of the original analog data tape is made that introduces a time lag, usually different for each track, by means of variable delay transmission lines. This delay process allows more exact registration between data channels. The skew between any two channels is usually reduced to less than 2 μ sec by this procedure. In theory, it is possible to correct for skew between channels by measurement of the differences in head alignment between the recording and playback equipment. In practice, however, the amount of time required for these measurements is prohibitive, and therefore the amount of delay to be applied to each channel during tape duplication is only approximated. Any further channel skew is calculated and corrected after the A-D conversion process is completed by the program AUTOCAL, discussed in Section 3.4. The corrected duplicate copy of the analog data tape is then used as input to the A-D processor for conversion to digital form.

Several regions along each scan must be digitized, because they contain information necessary to calibrate and cross-correlate the absolute data values in the video region, once these are in digital form. Built into the A-D converter is a resolution rate generator, which divides the region between two successive marker sync pulses into a specified number of regions each of which is digitized separately. However, it is not necessary to digitize the whole region between two sync pulses, since much of this is merely the scanner viewing the opaque inner wall of the instrument. The A-D converter may be gated manually such that only certain regions are digitized. The regions generally digitized are:

- (1) video, that region where the sensor views the scene below the aircraft
- (2) dark level, a portion of that region where the sensor views the dark interior of the scanner housing
- (3) sun sensor, that region where the sensor views the radiation passing through a diffuse opal glass plate on the top of the aircraft
- (4) calibration lamp, that region where the sensor views a radiance-transfer standard
- (5) cold plate, that region where the sensor views a source of cool, known temperature
- (6) hot plate, that region where the sensor views a source of known temperature, hotter than that of the cold plate

Figure 2 shows the time correlation between the analog data in a particular channel and the sync pulse channel, the digitizing gates, and the resolution rate generator pulses.

3.3. ANALOG-TO-DIGITAL CONVERSION WITH SMOOTHING

The actual digitizing procedure performed by the A-D converter is fairly simple:

- (1) A resolution element which has been selected to be digitized by means of the gate settings is read by the analog tape recorder; the data value of each channel is strobed into a 48-bit buffer and shifted right the appropriate number of places as designated by a data resolution switch containing the number of significant bits to be used per data value. The data resolution switch is ordinarily set manually to 9 bits per data element, resulting in an accuracy of 2^{-9} over the analog voltage range.
- (2) The completion of a 48-bit word causes the computer to transfer that word to storage in sequential, increasing addresses in core.
- (3) The CDC 1604 computer is signalled when the final resolution element for a particular scan line has been digitized and transferred to the CDC 1604 computer; a flag is sent to the 1604 that the block of data is complete. The computer then writes the digitized information onto magnetic tape, one scan line per digital tape record. (Appendix I describes the format of the digital tape.)

Flight altitudes of 2000 ft and above ensure some overlap in successive scan lines, i.e., several successive scan lines contain data from portions of the same region on the ground. This results in a quantity of analog data much larger than is necessary for reasonable data processing and evaluation. Thus, as the analog-digital conversion is taking place, an on-line computer program, designated A2F3, is being executed on the digital computer. This program performs a function called smoothing, or filtering, which has a twofold purpose:

- (1) The reduction of the number of scan lines by a factor between 3:1 and 16:1.
- (2) The filtering (or reduction) of instrumental noise present in the recording process.

This is accomplished by simple averaging of each data value over NSMOOTH lines, where NSMOOTH is the number of lines to be combined to form a single digitized line. If the distribution of instrumental noise associated with each scan line is assumed to be Gaussian, averaging cancels out much of this background.

The actual computational procedure uses the following areas within the computer:

- (1) Input buffer of NWORDS locations where

$$\text{NWORDS} = (\text{NSS} * \text{NCHAN} + 4) / 5$$

$$\text{NSS} = \text{the number of resolution elements per scan line}$$

$$\text{NCHAN} = \text{the number of channels per resolution element}$$
- (2) Scratch buffer consisting of two memory locations, used for intermediate, partial unpacking

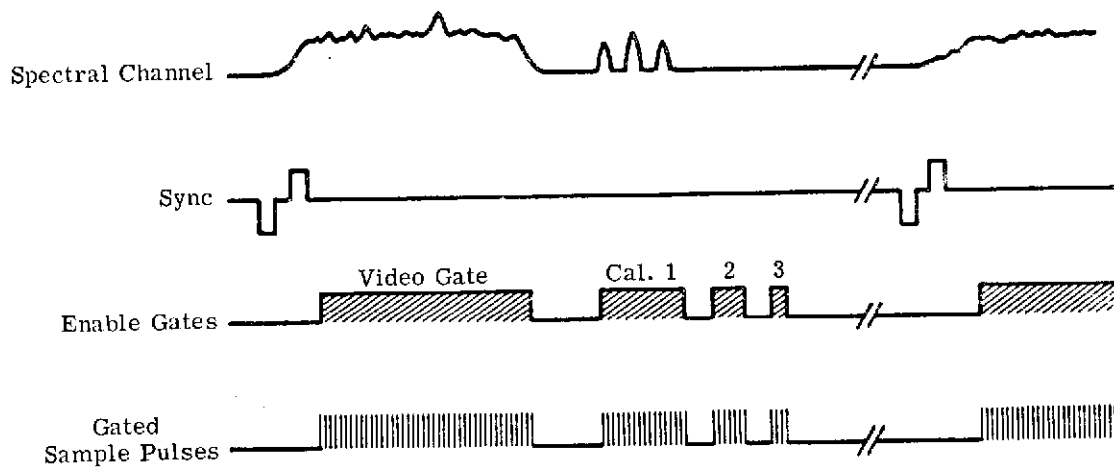
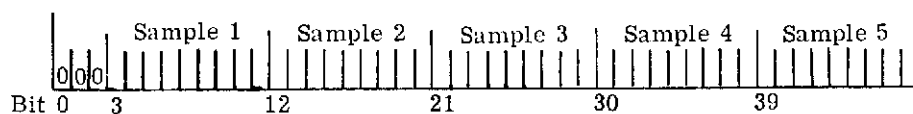


FIGURE 2. SCAN LINE WITH ENABLE GATES AND GATED SAMPLE PULSES

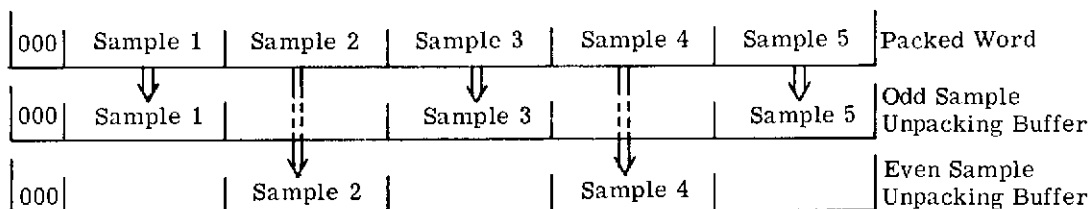
- (3) Two averaging buffers consisting of NWORDS memory locations, each of which is used to accumulate the data value averages
- (4) Output buffer of NWORDS memory locations, used for final repacking and output to magnetic tape

A memory location on the CDC 1604-B computer is composed of 48-bits; as scan lines are converted from analog to digital form they are directed to the computer memory and stored, 5 data values of 9 bits per word, with 3 bits unused (Fig. 3a). These data values are in 2's complement notation. Complement notation is described in Appendix I. The computational procedure used is:

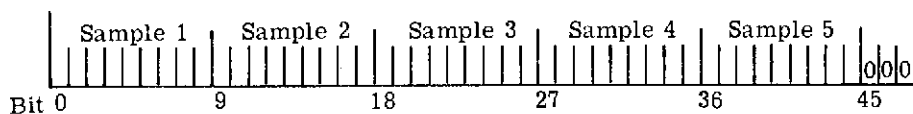
- (1) The first of NSMOOTH scan lines is read into the input buffer. Locations 1 through NWORDS are successively subjected to the following procedure:
 - (a) Each of the 5 data values in location N (where $1 \leq N \leq \text{NWORDS}$) is biased upward by complementing its sign bit.
 - (b) Each value is partially unpacked into one of two locations called partial unpacking buffers, the even samples into one location, the odd into the other. (See Fig. 3b.)
 - (c) If $\text{NSMOOTH} > 8$, the data values are each divided by 2 and the remainder discarded, with an accompanying loss of significance.
 - (d) The contents of these two locations are then stored in one of two (even or odd) averaging buffers.
- (2) Scan lines 2 through NSMOOTH are then successively read in, with steps 1a through 1d applied to each location of each scan line; however, step 1d is now modified such that the partial unpacking buffers are not stored in the averaging buffers, but are arithmetically added to them.
- (3) After scan line NSMOOTH has been processed, the averages are truncated to 9 bits per data value and multiplied by $1/\text{NSMOOTH}$, where this multiplier has been previously truncated to 9 bits. The product is then truncated to 9 bits.
- (4) The odd and even data values are repacked into the output buffer in a slightly different format (Fig. 3c), the bias removed by recomplementing the leftmost bit of each data value, and the data values converted to 1's complement by adding +1 to all negative data values.
- (5) Finally, the output buffer is written in ADTEST2 format (see Appendix II) on the output magnetic tape.



(a) A-D Generated CDC 1604B Data Word



(b) Partially Unpacked Data Words



(c) Packed Multispectral Data Output Word

FIGURE 3. PACKING OF MULTISPECTRAL DATA OUTPUT WORD

This procedure results in no loss of significance for smooths of 4- and 8-to-1; however, smooths of 9- through 16-to-1 initially lose 1 bit of significance, and smooths other than 4-, 8-, or 16-to-1 lose 1 bit of significance at the end of averaging.

3.4. AUTOCAL—EVALUATION OF CHANNEL SKEW AND SCAN LINE SLEW

After the A-D conversion process is completed, the resultant digital tape is used as input to a CDC 1604-B computer program, called AUTOCAL, which has a three-fold purpose:

- (1) Calculation of an average scan line, corrected for slew effects (spatial misregistration) between scan lines
- (2) From the calculated average scan line, a determination of the amount of skew present between channels, based on the position of the sun sensor peak in each channel (see Section 3.2)
- (3) Calculation of a dark level value for each channel of data

THE ALGORITHM SEQUENCE

The user identifies the channel (ICHAN) to be used as a reference in the calculation of skews for the other channels, and the boundaries C1 and C2 defining the correlation region used in both the deslewing and deskewing procedures. The program individually processes successive scan lines by editing noise spikes from the data, correcting for slew, and accumulating the sum line.

SPIKE NOISE EDITING

Data differences—i.e., the differences between the data values in each channel for successive resolution elements along the scan line—are computed. A noise spike is characterized by two very large differences of opposite sign enclosed by two ordinary differences. If the minimum of the two large differences is greater than 5 times the maximum of its two neighboring differences and also greater than a high estimate of the expected difference, DEL(J), a noise spike is identified, and the data value then replaced by the average of the preceding and succeeding values in the same channel. DEL(J) is obtained from the first line by ranking the absolute differences, and then multiplying the 80-th percentile difference by 2. DEL(J) is included to cover the case in which both the neighboring differences are very small or even zero. A steeply rising or falling signal is not identified as a noise spike if the neighboring pulses are much the same size and if the middle differences are of the same sign.

SLEW CORRECTION

Assuming that the scan lines consist of N resolution elements and NCHAN channels per resolution element, data differences (as defined previously) in channel ICHAN from the first data line between points C1 and C2 are computed and stored in a vector Y. Channel ICHAN is defined as the reference channel; C1 and C2 have been previously defined; and $C1 < C2 \leq N$.

Thus Y is a vector containing elements Y_1 through Y_C where $C = C_2 - C_1 + 1$. Then a set of sums, S_i , is formed such that

$$S_i = \sum_{k=j}^{j+C+1} D_{k,ICHAN} \cdot Y_{k-j+1}$$

where $i = 1, N - C + 1$ and where $D_{k,ICHAN}$ is defined as the data difference between the k -th and $(k + 1)$ -st resolution element in channel $ICHAN$ for the second scan line. Since maximum correlation occurs when this sum is a maximum, the data line is shifted in such a way as to align this region of maximum correlation with the region from C_1 to C_2 on the first scan line. (If there were no slew, the region of maximum correlation would occur in the C_1 to C_2 region for each scan line.)

After all the scan lines have been processed, the program computes the average line in each channel by dividing each element of the sum line by the number of lines processed. This average line is used for the calculation of the skew of each channel. The standards are points C_1 to C_2 on the $ICHAN$ average line. Each of the other lines is tested to find the region in which the length $C_2 - C_1 + 1$ correlates most closely with the standard. The method of correlation is, as before, the sum of products of differences, but instead of locating a local maximum, all possible sums with lags of -10 to $+10$ are tested and an absolute maximum obtained. The result of the correlation is output as a set of integers showing the number of points by which each channel precedes (negative) or follows (positive) the reference channel. Also, fractional slews for each channel are estimated by passing a parabola through the biggest sum and the two neighboring sums. The location of the base of the parabola determines the fractional slew, which differs from the integer slew by no more than 0.5 , a figure achieved when two adjoining sums are equal and maximal.

Then a sum line, $A_{i,j}$, is cumulated from the first data line and the shifted data line, such that

$$A_{i,j} = \sum_{i=1}^N \sum_{j=1}^{NCHAN} (V_{1,i,j} + V_{2,i,j})$$

where V_1 is the first scan line and V_2 is the second scan line.

Processing of this kind proceeds with successive lines (all lines being slew-corrected with reference to the first scan line). Each new shifted scan line is cumulated with the sum line A , such that

$$A_{i,j} = \sum_{i=1}^N \sum_{j=1}^{NCHAN} (A_{i,j} + V_{m,i,j})$$

where V_m represents the m -th data line after shifting for slew correction.

Finally, a dark level for each channel is computed from the average line. This dark level for channel j is defined as

$$DARK_j = MAX(S_i)$$

$$\text{where } S_i = \sum_{k=i}^{i+7} A_{i,j} \quad \text{for } i = 1 \text{ to } N - 7$$

Each value of the average line is subtracted from this DARK array, generating a corrected average line C , such that

$$C_{i,j} = DARK_j - A_{i,j} \quad \text{where } i = 1 \text{ to } N \\ \text{and } j = 1 \text{ to } NCHAN$$

The output from AUTOCAL provides information which may be used to provide further channel registration corrections and better boundaries for calibration regions, as inputs to the next stage of digital processing: clamping, scaling, and deskew (namely, the program CSD).

4

CLAMPING, SCALING, DESKEWING

Any or all of three types of corrections to the data contained in each scan line may be needed after an analog tape has been digitized:

- (1) dark level correction
- (2) multiplicative scaling based on one of the calibration sources
- (3) deskewing of the channels to correct spatial misalignment

Dark level correction and calibration scaling both employ data values contained in each scan line to calculate the correction factors. The deskew correction factors obtained from the AUTOCAL program are now used as input to this correction program, called CSD.

The user must supply the following correction data to the program CSD for the generation of a new data tape:

- (1) The line numbers of the scans to be corrected and the point numbers which represent the video portion of these scans

- (2) N skew values where N is the number of data channels on the tape, and the reference channel is assumed to have a skew of zero
- (3) For clamping (dark level correction), either
 - (a) a clamp region—i.e., line and point numbers of a dark level region, from which a dark level value for each channel is calculated
 - (b) actual clamp levels (dark levels) to be used
- (4) For scaling
 - (a) the point numbers of the region for the calibration signal of each line
 - (b) the sampling region line numbers if time channel sampling (Section 3.2) is to be used
 - (c) a smoothing constant representing the number of lines over which the peak in the calibration region is filtered by means of an exponential-type function
 - (d) multiplicative scale factors (optional)

4.1. CLAMPING

The user either specifies a clamping region to set the dark levels of each individual scan line, or supplies a constant dark level value for each channel.

If the dynamic dark level correction method is selected, the points specified by the user within the clamping region are averaged channel by channel, and an array DARK(I), I = 1, NCHAN is generated for each line. If dynamic dark level correction is not used, DARK(I), I = 1, NCHAN remains constant throughout the run.

The correction is applied such that for DATA(N, J) where N = 1, NSS and J = 1 NCHAN is modified to read $DATA(N, J) = DATA(N, J) - DARK(J)$.

4.2. SCALING

Three scaling options are available:

- (1) dynamic scaling with a time channel criterion
- (2) dynamic scaling without a time channel
- (3) multiplicative constant scaling (user-specified)

The use of any of these options alone is possible, as is a combination of either 1 and 3 or 2 and 3.

If a time channel (Section 3.2) is available, the procedure is as follows:

- (1) A region is specified by the user, and the range of values for that region is calculated.
- (2) Each line of the entire scene to be processed is then tested to see if the value of the time channel for the particular line is within the range of time channel values for the region specified by the user in (1) above. If the value for the particular scan line is within this range, dynamic scale factors are calculated as for scaling option (2) above and applied to the scan line. If the time channel value for the scan line is outside the acceptable range (as defined by the user-specified region), the scale factors calculated for the previous line are applied to the current scan line.

Dynamic scale factors are calculated as follows:

- (1) A value is calculated for each resolution element of the scaling region as specified by the user; this value is the sum of all the channel values for the resolution element.
- (2) The maximum of these values is found, and data values in all channels at this peak element are filtered, by means of an exponential function to avoid noise spike problems, with the previous LSMOOTH lines where LSMOOTH is the number of user-specified lines for the smoothing process (default = 60).
- (3) The smoothed values for each channel are then used to divide the data in the video region.

4.3. DESKEWING

For NSS resolution elements per scan line in the video region and NCHAN channels per resolution element, the user must supply NCHAN values into an array ISKEW(I) where $I = 1, 10$ ($1 \rightarrow I \rightarrow 10$ in increments of 1). The variable MINSKEW is set equal to the minimum value of all of the NCHAN skew values in the ISKEW array.

Then the ISKEW array is modified such that MINSKEW is subtracted from all of the NCHAN values of the ISKEW array.

When $DATA(N, I)$ (the value of the J -th channel for the N -th resolution element where $J = 1, NCHAN$ and $N = 1, NSS$) is stored for scaling, $DATA(N, J)$ is not returned, but replaced by the value $DATA[N + ISKEW(J), J]$.

5

DATA DISPLAY

Displays of the multispectral data are produced by means of the CDC 1604-B computer program GRAY2.

The user specifies a method by which the data values for an array called `LEVEL` are established. The `LEVEL` array contains data values which encompass the entire range of possible values for a given set of data (i.e., 0 - 511 for positive data, and -255 - 255 for bipolar data). These levels increase in value with increasing subscript, and a set of 2 characters is associated with a given level.

The following methods are used to perform level settings:

- (1) Manual method: The user feeds levels directly to the program. The program performs elementary tests to ensure that the level values are monotonically increasing and encompass the entire range of possible data values.
- (2) Automatic level set by random sampling: This method randomly selects points from a user-specified region (usually a subset of the region to be displayed) using a programmed random number generation routine. Then `NLEVEL` display levels (`NLEVEL` being a user-defined variable specifying the number of display levels) are set such that the distribution of the randomly-selected data values is uniform in each range delineated by the `LEVEL` values.
- (3) MINMAX method: A minimum and a maximum range value, `MIN` and `MAX` respectively, are user-specified, along with `NLEVEL`; levels are generated such that the difference between values of the `LEVEL` array is $\text{DIFF} = (\text{MAX} - \text{MIN})/\text{NLEVEL}$.

When the levels have been determined, association of each resolution element with one of the `NLEVEL` levels takes place in one of three modes.

Simple classification: The data value in the channel to be displayed for a given resolution element is compared to the i -th level where $i = 1, \text{NLEVEL}$. When the data value for the resolution element is greater than `LEVEL(I - 1)` but less than or equal to `LEVEL(I)`, the resolution element is assigned the set of symbols associated with the I -th level for printing.

Two-channel interval criterion classification: The level to which a particular resolution element is classified is selected in the same manner as for the simple classification method; however, the value of the resolution element in a second channel (a control channel) is then tested. If the value in the control channel is less than a user-supplied maximum value, the

resolution is assigned the I-th set of symbols for printing; however, if the value in the control channel is larger than the maximum allowed, the resolution is assigned a blank character for printing.

Two-channel level-equality criterion classification: The level to which a particular resolution element is classified is selected in the same manner as for the simple classification method; however, the value of the resolution element in a second channel (a control channel) is compared to the values of the array LEVEL. If the control channel value matches one of the LEVEL values, the resolution element is assigned the I-th set of symbols (where the element was classified as the I-th level initially); otherwise, the resolution is assigned a blank set of symbols for printing.

Each point of the scan line is processed in this manner (that is, associated with a particular level and thus with the particular set of print symbols characteristic of that level); then a computer printout is produced, such that one line of printed symbols bears a one-to-one correspondence to the data values for a given scan line. Figure 4 exemplifies the format of this visual multispectral data display. A discussion of recognition color mapping is contained in Section 7.

INTEGER RANGE		VOLTAGE RANGE		SYMBOL	ID
0	10	0.0000	10.0000	●	MCV
11	20	11.0000	20.0000	■	DBS
21	30	21.0000	30.0000	■	BRV
31	40	31.0000	40.0000	■	TRW
41	50	41.0000	50.0000	■	POV
51	60	51.0000	60.0000	■	PPV
61	70	61.0000	70.0000	■	WATER
71	80	71.0000	80.0000	■	MDS
81	90	81.0000	90.0000	■	LBS
91	511	91.0000	511.0000		

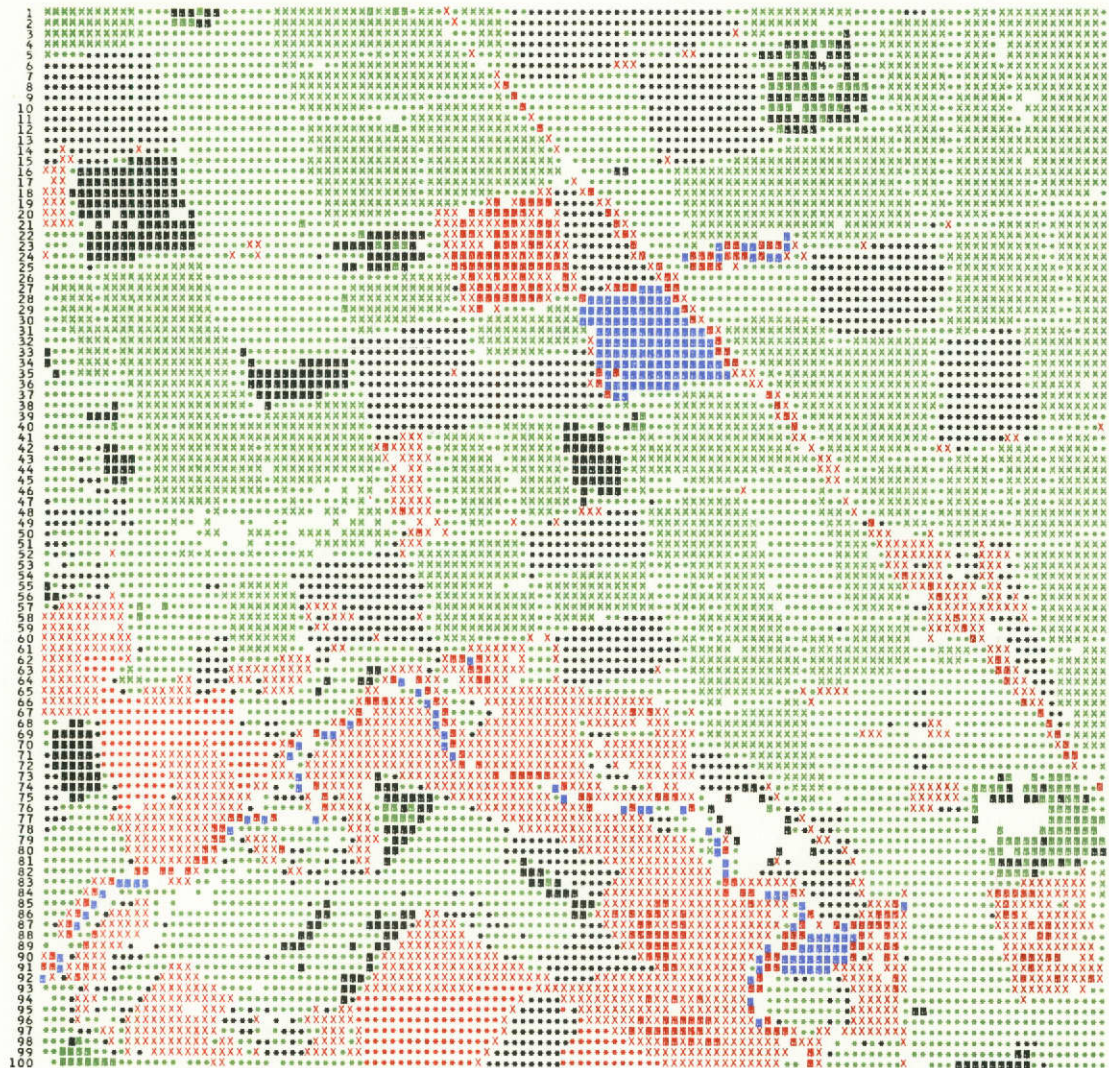
[illegible][illegible]

FIGURE 4. EXAMPLE RECOGNITION DISPLAY



6

SIGNATURE EXTRACTION

The end goal of multispectral data processing is determination of the object class of a given resolution element in the target area. This is accomplished by defining areas, called training sets, which are assumed to be homogeneous and representative of a particular object class. The data values in a training set region are combined such that the set of mean and covariance matrices, along with secondary statistical information, define what is called the signature of a particular training set. Once a representative number of training sets and their associated signatures have been obtained, classification procedures (the assignment of a given resolution element to a particular class, based on comparison of the resolution element's data values with the set of signatures) are instituted on the resolution elements of a particular subset of the data, called the scene, chosen by the user.

This section will now describe the algorithms implemented for the extraction of signatures by the CDC 1604-B program SIG1.

In the calculation of a given signature, the algorithm implemented in SIG1 is as follows:

- (a) All training sets must be assumed rectangular, and the user must supply scan line and point boundaries for each training set.
- (b) The mean value in each channel, a covariance matrix, the standard deviation from the mean in each channel, and a correlation matrix, are accumulated by processing the points within the training set individually.

Given a two-dimensional DATA array, where DATA(I, J) refers to the value of the J-th channel of the I-th point to be processed, the processing of n data points will result in:

$$\text{MEAN}(J) = \frac{1}{n} \sum_{I=1}^n \text{DATA}(I, J)$$

For the covariance matrix entries for channels J and K (where J and K are less than or equal to NCHAN), the total number of channels used to calculate the signature is

$$\text{COV}(J, K) = \frac{1}{n-1} \left[\sum_{I=1}^n \text{DATA}(I, J) * \text{DATA}(I, K) \right] - \text{MEAN}(J) * \text{MEAN}(K)$$

the standard deviations

$$\sigma_J = \sqrt{\text{COV}(J, J)}$$

and the correlation matrix

$$\text{COR}(J, K) = \frac{\text{COV}(J, K)}{\sigma_J \sigma_K}$$

- (c) The values calculated for each training set are based on n data points being used. The program edits data points within the training set region so that extreme points are not used in the signature calculation.

The editing procedure begins with the establishment of editing criteria. Each component of the data point is compared to the edit values. If any component is outside the acceptable range then the whole data point is rejected.

The establishment of editing criteria requires an estimate of the parameters of the distribution being used so that extreme points or points not from that distribution may be excluded. A criterion based on a sample median and sample quartiles is preferable to one based on a sample mean and sample standard deviation because extreme points have much less effect on the median and quartiles. With the assumption of a form for the underlying population distribution and a probability threshold for exclusion, one can set upper and lower bounds using the median and quartiles. An underlying multivariate normal population is assumed, so the median as an estimate of the mean and the quartiles can be used to estimate the standard deviation.

To use the normal (z) distribution, it is necessary to assume that the median is the population mean value and that the average quartile deviation from the median represents a value $z_Q = 0.6745$ (see Fig. 5). Thus,

$$z_Q = \left[\frac{1/2(X_{3/4} - X_{1/4})}{\sigma} \right] = \frac{X_Q}{\sigma}$$

so

$$\sigma = \left(\frac{X_Q}{0.6745} \right)$$

Next, a probability threshold criterion for excluding points is picked, based on the above assumptions. For SIG1, only one point in a thousand is to be rejected if the data meet the assumptions. This is equivalent to saying that the point should be rejected if

$$|z| = \left| \frac{X - \text{Median}}{\sigma} \right| > 3.2905 \quad (\text{See Fig. 6})$$

The SIG1 edit routine considers the individual channel values to be independent normal samples, computes bounds separately for each channel, and rejects an observation point if any one of its channel values exceeds its bounds. Thus, the probability of any one channel exceeding its bounds must be $(1/NC)$ times the criterion value (0.001), where NC is the number of channels. Thus, if $NC = 10$, so $P(|z| > z_T) = 0.0001$, then the criterion is $z_T = 3.89$. Since the threshold X values, X_T , are given by:

$$\begin{aligned} X_T(NC) &= \text{Median} \pm z_T(NC)\sigma \\ &= \text{Median} \pm \left[\frac{z_T(NC)}{z_Q} \right] X_Q \\ &= \text{Median} \pm \frac{z_T(NC)}{0.6745} X_Q \end{aligned}$$

We have, for $NC = 10$,

$$X_T(NC) = \text{Median} \pm 5.767 X_Q$$

Note that there is no dependence on the number of samples used, except indirectly in that the median and quartiles will more accurately estimate the true population parameters as more samples are used.

The algorithm employed to determine the editing criteria is as follows.

The first N data points in the area being processed are used to form the sample population. The value of N depends on the number of channels being used; N is given by the formula:

$$N = 2400/NC$$

also

$$0 < N \leq 600$$

Computer storage limitations account for the limit of an upper bound on N.

Then, for each channel, the median and quartile deviations are found. The editing bounds are calculated as:

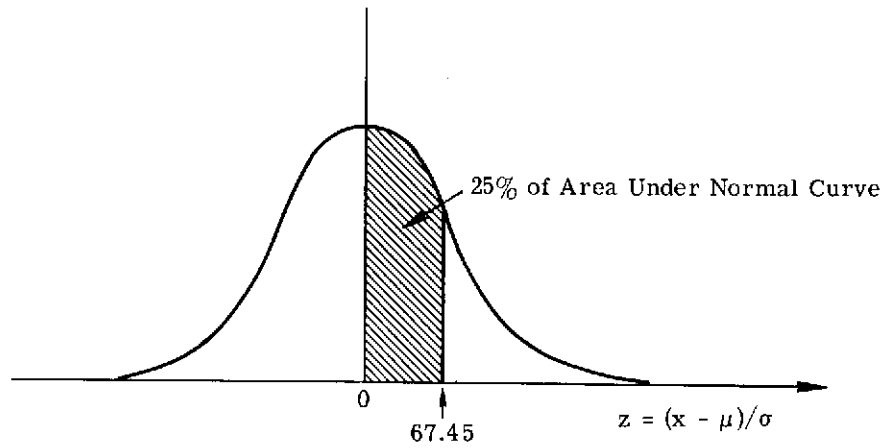


FIGURE 5. NORMAL DISTRIBUTION. Quartile deviation.

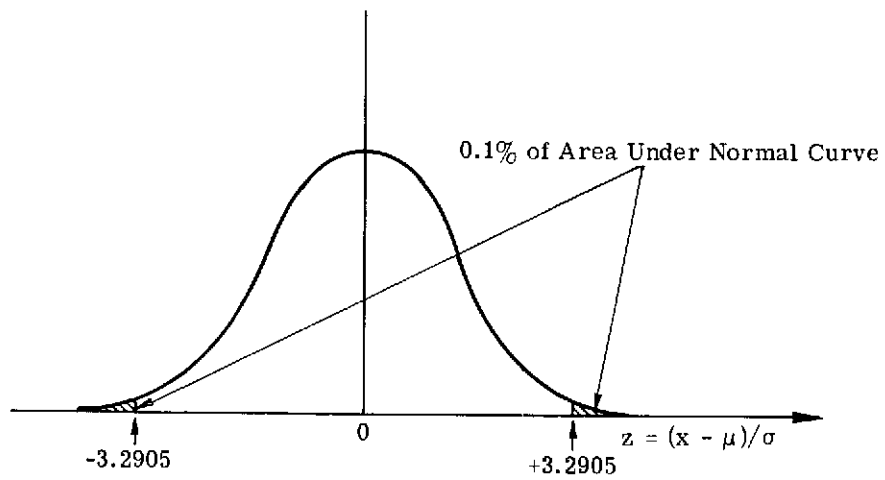


FIGURE 6. REJECTION CRITERIA

$$HI(I) = \text{MEDIAN} + QDEV * \text{EDFACTOR}(\text{NC})$$

$$LO(I) = \text{MEDIAN} - QDEV * \text{EDFACTOR}(\text{NC})$$

where $\text{EDFACTOR}(\text{NC})$ is the $\left[\frac{Z_T(\text{NC})}{0.06745} \right]$ term from before.

The values for EDFACTOR for channels 1 - 24 are:

CHAN	1	2	3	4	5	6	7	8
EDFCTR	4.88	5.16	5.32	5.43	5.51	5.58	5.64	5.69
CHAN	9	10	11	12	13	14	15	16
EDFCTR	5.73	5.77	5.80	5.83	5.86	5.89	5.91	5.94
CHAN	17	18	19	20	21	22	23	24
EDFCTR	5.96	5.98	5.99	6.01	6.03	6.05	6.06	6.08

Where $LO(J)$ and $HI(J)$ are the lower and upper editing limits, respectively, in the J -th channel, the individual points are not deleted if

$$LO(J) \leq \text{DATA}(I, J) \leq HI(J)$$

for all channels, where $\text{DATA}(I, J)$ is the value of the J -th channel for the I -th resolution element within the training set.

7

RECOGNITION PROCESSING

Once signatures have been extracted from a scene of multispectral data, it is then possible to attempt to associate a given resolution element with one of the known signatures. This process is called classification of multispectral scanner data, and is performed by the CDC 1604-B program EXPMAP.

With M signatures available for a given scene of data to be classified, a value Q is computed for each of the M signatures for each resolution element to be processed, where

$$Q_i = (X - \mu_i)^T R_i^{-1} (X - \mu_i)$$

where X = the vector containing the data values in all the channels used

μ_i = the mean vector for the i -th signature

R_i = the covariance matrix for the i -th signature

The resolution element in question is said to be classified as the j -th signature class, when Z_j , where $Z_j = Q_j + \log |R_j|$, is greater than the Z_k computed for any of the other signatures.

Initially, before any classification processing takes place, the user must input an array S such that the k -th signature is associated with the element S_k , where $0 \leq S_k < 511$ for all k . As the i -th resolution element is classified as belonging to signature k , a two-channel output scan line is being constructed such that the first channel contains the value S_k and the second channel the value EXP_i , where

$$EXP_i = Q_{ij} * 5.12 + 0.5 \quad \text{when } Q_{ij} \leq 99.6$$

and Q_{ij} is the Q value computed above for the i -th resolution element and the j -th signature class. If $Q_{ij} > 99.6$, $EXP_i = 511$.

Thus the second channel can be used as a criterion denoting how well the classification procedure worked for each resolution element.

8

RECOGNITION MAPPING

After the classification or recognition procedure has been completed, display of the results in a useful manner is necessary so that signature selection and classification techniques can be evaluated. Since the output from the EXPMAP classification program is another digital tape of the standard format (described in Appendix II), the display of the data can be accomplished by the program GRAY2 (Section 5).

The two-channel modes described in Section 5 are used primarily for displaying recognition output tapes. The two-channel interval criterion mode is used to display only those points whose classification was reasonably certain, based on the value stored in the second channel by EXPMAP (Section 7). It is possible with this feature to generate displays with different cut-off values for this second channel, and therefore with different error bounds.

The two-channel level-equality criterion mode allows the user to evaluate the probability of correct classification over the scene for a given signature class. Consequently, the user might find that at the edges of a particular region the probability of correct classification was poorer than at the center of the region.

The visual display of recognition output is extremely helpful in evaluating the recognition process, since it allows a large amount of data display in a relatively small amount of output in a form easily scanned by the user and in which gross features become more prominent than individual resolution elements.

Figure 4 is an example of a color-coded recognition output visual display.



Appendix I COMPLEMENT NOTATION

One's Complement

The one's complement of a binary number is defined as that number which results by replacing each digit 1 in the original number with the digit zero and replacing each zero with a 1. For example, the one's complement of the binary number 101 is the binary number 010. Note that the sum of a binary number and its one's complement is the binary number composed of all ones (e.g., $101 + 010 = 111$).

Two's Complement

The two's complement of a number is defined as the one's complement increased by 1. Note that if an n -digit binary number is added to an m -digit binary number (where $n \geq m$), any carry to the $(n + 1)$ st position is ignored. Thus a binary number added to its two's complement is a number consisting of all zeroes ($101 + 011 = 000$).

Appendix II

DIGITAL TAPE FORMAT

One set of scan lines as recorded on analog tape results in a set of digital records, terminated by an end-of-file, as follows:

- (1) The first record, 2592 bits (54 CDC 1604-B words) in length, is called the title record and contains information pertinent to the data set which will follow. The content of this title record is:

1	word	Fortran code word (17777777777777 ₈)
12	words	BCD title of data set
1	word	number of resolution elements per scan line (NSS)
1	word	beginning angle from nadir (BANG)
1	word	angular increment (DANG)
1	word	number of channels (NCHAN)
1	word	conversion factor (CONV)
1	word	data positive flag (IPOS)
1	word	data packed flag (IPACK)
24	words	descriptive information in BCD
10	words	unused

- (2) The rest are data records, containing one scan line of data per record. The number of CDC 1604-B words per record is

$$NWORDS = \left[\frac{(NSS \times NCHAN) - 1}{5} \right] + 1$$

Each word contains five 9-bit data values (Fig. 3), such that the data values representing channels 1 through NCHAN for the i-th resolution element are contiguous.

- (3) The data set is terminated by a standard end-of-file mark.

Appendix III PROGRAM LISTINGS

```

PROGRAM SIG1
C  NEEDS PROCESS, UNPACK3, POSDEF
C  VERSION 1.0 (PROGRAMMERS W.L.BROWN + D.ZUK, 12/71)
***WARNING*** SURR. POSDEF USES 2*NC**2 +NC + 3 SPACES IN ERASABLE
***FOR THIS REASON, THE 'DATA' ARRAY SHOULD BE KEPT FIRST IN COMMON

*****'SIG1' IS A SPECIAL VERSION OF 'IMPROVE' WHICH ENABLES ONE TO CAL-
C  CULATE STATISTICS FOR SEQUENTIAL, CONTIGUOUS TRAINING AREAS OF
C  EQUAL SIZE AS WELL AS 'NORMAL' TYPES OF TRAINING SETS
C  'SIG1' IS A GENERAL SIGNATURE-CALCULATING PROGRAM WHICH REPLACES
C  'GREGCAST' ... A FORM OF HISTOGRAM IS INCLUDED IN THE OUTPUT
C  DOES NOT ALLOW TRANSFORMATIONS

C  PACKED DATA TAPE ON 3, BCD OUTPUT ON 4
C  PUNCHED OUTPUT CAN BE WRITTEN ON ANY TAPE DRIVE OR ON '0' TO DISCARD
C  BCD OUTPUT OF TAPE CHANNEL ICP ON 5 (ICP=0 TO OMIT OUTPUT ON 5)

C  INPUT MEDIUM IS T, F, 1, 2, 6, 7 OR 8
C  'INPUT MEDIUM' IS REQUESTED PRIOR TO READING IN PROGRAM CONSTANTS
C  AND THEN AGAIN PRIOR TO READING IN LINE, POINT DESIGNATIONS... THIS
C  ENABLES ONE TO USE THE SAME CONTROL CARDS AS FOR 'HIST1' AFTER
C  SECOND ASSIGNMENT OF INPUT MEDIUM IS MADE

C  UP TO 13 CHANNELS, 6000 SAMPLES/SCANLINE

C  SENSE SWITCH 1 TO TURN OFF EDITING
C  SENSE SWITCH 2 IF RUN CONTINUES OVER 2 OR MORE FILES
C  SENSE SWITCH 3 FOR OPTION TO CHANGE TITLES

C  FOR CHANNEL NOS. INSTEAD OF WAVELENGTH BANDS IN OUTPUT, ANSWER
C  'SPECTROMETER CHANNELS=' WITH 0 OR WITH '15'
C  SPECTROMETER IDENTIFICATIONS ARE...
C
C      SPECTROMETER CHANNEL      BANDPASS
C      -----
C          1          .43 - .47
C          2          .47 - .49
C          3          .49 - .51
C          4          .51 - .53
C          5          .53 - .56
C          6          .56 - .59
C          7          .59 - .63
C          8          .63 - .67
C          9          .70 - .90
C         10         1.00 - 1.40
C         11         1.50 - 1.80
C         12         2.00 - 2.60
C         13         9.30 - 11.7
C         15         NO LABEL

C  (BANDS ARE FOR M-7 SCANNER. CF. E. WORK 7/6/71 MEMO)
C  NC = NO. OF CHANNELS IN SUBSET. (OPTIONAL)
C  SUBSETS ARE READ BY (2013)
C  ICP = TAPE CHANNEL TO BE PLOTTED ON TAPE UNIT 5
C  ICP = 0 BYPASSES WRITING ON TAPE 5

C  NTERM = 1 FOR PLOTS OF TEMPERATURE (DEG. C.) VS. WAVELENGTH
C  NTERM = 0 FOR VOLTS VS. WAVELENGTH
C  NTERM = -1 FOR REFLECTANCE VS. WAVELENGTH

```

```

C  NTERM = -2 FOR RADIANCE VS. WAVELENGTH (RADIANCE UNITS = WATT/SQ.CM./
C  CM./STER) X 10**ICST)
C  ICST = SCALING CONSTANT FOR THE PLOTS

C  LINE NO. = NSA, NSB, KS, NA, NB, KP, ID1, ID2 (615, 2A8)
C  LINE NO. .LE. 0 TO GET TO 'NO. OF FILES TO SKIP = ' QUESTION
C  NC. OF FILES TO SKIP = -0 TO CHANGE CHANNEL SUBSET
C  NSB = NO. OF LAST LINE OF INTEREST (NSB=-1 TO GO TO EOF)
C  NSK = NC. OF SCANLINES IN EACH STATISTICAL SET
C  NSK = 0 FOR STANDARD 'IMPROVE'-TYPE RUN
C  'ID1' CAN BE STANDARD AS IDENTIFICATION OR CAN BE THE TIME OF DAY
C  (FORMAT = XXXX.X) CORRESPONDING TO FIRST AREA
C  LEAVE ID1, ID2 BLANK TO SIGNAL IRREGULARLY SHAPED AREAS, PUTTING IN
C  ID ONLY WITH THE LAST LINE + POINT DESIGNATIONS
C  ID2 CAN BE STANDARD AS IDENTIFICATION OR CAN BE 'DELTA TIME' (F6.3)
C  DELTA TIME = TIME INTERVAL BETWEEN AREAS (NEEDS TO BE ENTERED ONLY
C  ONCE AFTER PROGRAM IS CALLED UNLESS IT IS DESIRED TO CHANGE IT)

```

```

      DIMENSION A(19), FLL(24), LAB(5), LABX(3), LBL(5,3), PCILE(21),
1     SCAN(200), T(12)
      COMMON DATA(6000)
      COMMON FMT(19), IC(13), IT(9), LABEL(30,3), LINE(108), MINE(100),
1     P(39), PP(3), Q(39), QP(3)
      COMMON B(13,13), STDEV(13), SUM(13), NGOOD(13), FN(13,13)
      COMMON BOT(13), CORR(13), ISCN(21,13), DATUM(13), FLHI(13),
1     FLLQ(13), ICODE(13), ICP, NC, R, S, TOP(13), V(200,13)
      COMMON RS(5), BANG2, DANG2, CC, NF, NR, MR, L80, ID1, ID2,
1     BANG, DANG
      COMMON L90, KEY, NPTS, NLINES, NSA, NSB, KS, NA, NB, KP, IS,
1     TITLE(12), TAG1(12), TAG2(12), NSS, NCHAN, KR, CONV, IPDS,
2     IPACK, INT
      EQUIVALENCE (LINE, SCAN), (T, TITLE)
      INTEGER DATA, FMT, Q, QP, R, REPLY, S, T, TAG1, TAG2, TITLE, U, W
      REAL ISCN
      LOC (TEST = 70)
      DATA A=10., 8., 6., 4., 3., 2., 1., .8, .6, .4, .3, .2, .1,
1     .08,.06,.04,.03,.02,.01)
      DATA(FLL = .41, .43, .455, .47, .485, .50, .52, .55, .58,
1     1.63, .68, .74, .85)
      DATA(LAB = 8H*.01* , 8H*.1 * , 8H*1.0* , 8H*10.* , 8H*100* )
      DATA(LABX= 8H 12X , 8H 19X , 8H 27X )
      DATA(LBL = 8H*2* 7X,8H*3* 6X,8H*4* 7X,8H*6* 6X,8H*8* 4X,
1     8H*2* 12X,8H*3* 9X,8H*4* 12X,8H*6* 8X,8H*8* 7X,
2     8H*2* 16X,8H*3* 12X,8H*4* 16X,8H*6* 12X,8H*8* 9X)
      DATA(PCILE(1) = 0.,5.,10.,15.,20.,25.,30.,35.,40.,45.,50.,55.,
1     60.,65.,70.,75.,80.,85.,90.,95.,100.)
      DATA(SECOND = 1.)
      PARTMAP

```

```

      SLJ(1)

600  SLJ(*)
      LDA(NC)          SUB(1)      .INITIALIZE FOR NEXT REGION
      WRITE(4,105)      AJP(602)

602  DO 610 JK=1,19
610  FMT(JK) = 8H

      CALL ZERO(B, FN(169))

```



```

      IP = 0
      IRREG = 0
      NFIRST = 0
      SLJ(600)

C      SET UP TO CONSTRUCT LABELS FOR COMPUTER PLOTS
1    DO 15 IL=1,3
      LABEL(1,IL) = LAB(1)

      DO 8 JL=3,7
6    LABEL(JL,IL) = LBL(JL-2,IL)
      LABEL(8,IL) = LAB(2)

      DO 4 JL=10,14
4    LABEL(JL,IL) = LBL(JL-9,IL)
      LABEL(15,IL) = LAB(3)

      DO 5 JL=17,21
5    LABEL(JL,IL) = LBL(JL-16,IL)
      LABEL(22,IL) = LAB(4)

      DO 6 JL=24,28
6    LABEL(JL,IL) = LBL(JL-23,IL)

      DO 7 JL=30,37
7    LABEL(JL,IL) = LABX(IL)

15   LABEL(29,IL) = LAB(5)
      SLJ4(600).          INITIALIZE FOR FIRST REGION
      DELTA = 0.
      K = 1
      L = 0
      MSA = 0
      N3 = 3
      NSR = 0

101  FORMAT(6I5)
102  FORMAT(5E15.8)
103  FORMAT(20I3)
104  FORMAT(A1)
105  FORMAT(1H1)
106  FORMAT(12A8)
109  FORMAT(1H0)
118  FORMAT(1H1, 12A8, / 1H , F4.2, 3H TO F6.2, 12H MICRON BAND /)
121  FORMAT(/ 12X 1H0 10X 2H1. 10X 2H2. 10X 2H3. 10X 2H4. 10X 2H5.
1    10X 2H6. 10X 2H7. 11X 11HMEAN(STDEV) /)
123  FORMAT(1H 14A8)
142  FORMAT(F8.1)
144  FORMAT(1H0 2X 14HCHANNEL NO. = 10X 13I7)
145  FORMAT(8H LINES 15, 5H THRU 15, 8H EVERY 13, 10H, POINTS 14,
1    5H THRU 15, 8H EVERY 13>(* F6.2 * DEGREES TO * F6.2 * DEGREES*
2    * )*/)

      LDA(SECOND)      AJP(12A)
      SECOND = 0.

      WRITE(9,100)
100  FORMAT(3H**** JS1 ON TO TURN OFF EDITING* /
1    3H**** JS2 ON TO CONTINUE SEGMENTS ACROSS EOF* /
2    3H**** JS3 ON TO ALLOW TITLE CHANGES* /)

```

```

12A WRITE(9,107) 358, 528
107  FORMAT(* PUNCHED OUTPUT ON * 2R1)
    READ(9,104) W2
    W = 0
    IF DATA CONVERSION ERROR GO TO 12
    DECODE(1, 101, W2) W

12  SLJ4(700).                SELECT INPUT MEDIUM
    NFIRST = 1
    WRITE(S,108)
108  FORMAT(*      NC  NSK  ICST  ICP  NTHERM (5I5) * /)
    READ(R,101) NC, NS3, ICST, ICP, NTHERM
    NC = MIN0(NC, 13)
    ITHERM = NTHERM + 3
    NTHERM = 0
    LDA(ITHERM)      SUB(4)      AJP1(14)
    NTHERM = 1

14  CALL PROCESS(DATA,1,0,N3,R)
    IF(KEY.EQ.5) 12, 13

13  ISBAD = 7778
    LDA(IPCS)      AJP1(L+2)
    LAC(0)          STA(ISBAD)
    BADPT = FLGAT(ISBAD)/CONV
    LDA(NCHAN)      INA(-2)      AJP2(94)
    NC = 1
    NC1 = 1
    NC2 = 1

94  LDA(NC)        AJP1(11)
    NC = MIN0(13, NCHAN)

11  NC3 = MAX0(1, NC/2)
    NC4 = MIN0(NC, NC3+1)
    NC1 = ICODE(NC3)
    NC2 = ICODE(NC4)

C  NC1 + NC2 ARE USED IN EDITING,.. IF THE SIGNALS IN BOTH OF THESE
C  USUALLY STRONG CHANNELS ARE 'BAD', POINT WILL BE DISCARDED

    WRITE(S,110) NC
110  FORMAT(* SPECTROMETER CHANNELS(*I2*I3)= *)
    READ(R,103) (IC(I),I=1,NC)
    FLLO(1) = 0.

    DO 10 I=1,NC
        ICODE(I) = 1
        LDA1(IC)      AJP(10)
        ITMP = (16*ITMP)/15
        IC(I) = (18*ITMP)/17
        FLLO(I) = FLL(IC(I))
        FLHI(I) = FLL(IC(I)+1)
10  CONTINUE

    LDA(NC)          SUB(NCHAN)  AJP(13A)
    WRITE(S,111) NC
111  FORMAT(* SUBSET OF CHANNELS(*I2*I3)= *)
    READ(R,103) (ICODE(I), I=1,NC)

13A SLJ4(700).                SELECT INPUT MEDIUM

```

```

2  CALL PROCESS (DATA,2)
    INT = 1
    ANGA = BANG2
    ANGB = ANGA + (FLOAT(NPTS)-1.)*DANG2
    LDA(KEY)    SUB(5)    AJP(12)

    IF(ID1 .EQ. 1H ) 21, 20
20  IF DATA CONVERSION ERROR GO TO 21
    DECODE(1,142,ID1) TEMP
    DECODE(1,142,ID2) DEL
    LDA(DEL)    AJP(21)
    ID2 = 1H
    DELTA = CEL
21  NSK = NS3
    AJP1(22)
    NSF = NSB
    NSK = NSB - NSA + 1
22  IF(ID1 .EQ. 1H ) 23, 24
23  IRREG = 1
    WRITE(4,145) NSA, NSF, KS, NA, NB, KP, ANGA, ANGB
    SLJ(25)
24  IRREG = 0
25  SLJ2(26)    SLJ(27)
26  MSA = NSR
    NSR = 0
27  NSF = NSA + NSK - 1 - MSA
    MSA = 0
    EDIT = 10.
    SLJ1(29)
    EDIT = 0.
29  NFIRST = 1

    SLJ3(39)    SLJ(37)

39  WRITE(9, 117)
117  FORMAT(* NEW TITLE * /)
40  READ(9, 106) IT
    IF(IT(1) .EQ. 3HYES) 40, 41
41  IF(IT(1) .EQ. 2HND) 37, 42

42  DO 43 I=1,9
43  TITLE(I) = IT(I)

37  IF(NC .EQ. 1 .AND. NFIRST .GT. 0) 44, 45
44  WRITE(4,118) TITLE, FLLO(1), FLHI(1)
    SCALE = 12.
    FTHERM = 1.
    OFFSET = 1.

45  MGCD = ((NSF-NSA)/KS+1)*((NB-NA)/KP+1)

3  CALL PROCESS (DATA,3)

    LL = NCHAN*(NA-1)

DO 49 ISS=NA,NB,KP
    LDA(IP)    SUB(200)    AJP3(47)

```

```

DO 46 J=1,NC
  L = LL + ICODE(J)
  LDA4(DATA)      AJP1(46A)
  LDA(ISBAD)      STA4(DATA)
46A DATUM(J) = FLOAT(DATA(L))/CONV
46  CONTINUE

  LL = LL+NCHAN*KP
  SLJ4(200).
  SLJ(49)
TEST POINTS AND CUMULATE TOTALS

47  IP = IP + 1

DO 48 J=1,NC
  L = LL + ICODE(J)
  LDA4(DATA)      AJP1(48A)
  LDA(ISBAD)      STA4(DATA)
48A V(IP,J) = FLOAT(DATA(L))/CONV
48  CONTINUE

  LL = LL+NCHAN*KP
  IF(IP .EQ. 200) 51, 49
51  SLJ4(300).
GET BOUNDS FOR COLLECTED POINTS

49  CONTINUE

  IF(NSF-IS) 52, 52, 50
50  IF(KEY-3) 3, 52, 90

52  NGCOD = NGCOD + MGOOD
  LDA(IRREG)      AJP1(2)
  LDA(IP)         SUB(200)
  SLJ4(300).
  AJP2(53)
GET BOUNDS FOR COLLECTED POINTS

53  DO 54 I=1,NC
54  SUM(I) = SUM(I)/FN(I,I)

DO 55 I=1,NC

DO 55 J=I,NC
  B(I,J) = B(I,J)/FN(I,J) - SUM(I)*SUM(J)
55  B(J,I) = B(I,J)

  LDA(NC)         SUB(1)      AJP1(60)
  LDA(NFIRST)     AJP(56)
  LDA(NTHERM)     AJP(56)

WRITE(4,120)
120 FORMAT(/ 10X 3H-40 9X 3H-30 9X 3H-20 9X 3H-10 10X 1H0 10X 2H10
1 10X 2H20 10X 2H30 10X 3H40 11HMEAN(STDEV) /)
  OFFSET = 61.
  FTHERM = 10.
56  P(2) = SUM(1)
  B(1,1) = AMAX1(B(1,1), 0.)
  STDEV(1) = SQRT(B(1,1))
  P(1) = P(2) - STDEV(1)
  P(3) = P(2) + STDEV(1)
  FSUM = SUM(1)*FTHERM
  FSTDV = STDEV(1)*FTHERM

```

```

DO 57 J=1,3
  Q(J) = P(J)*SCALE + OFFSET
  Q(J) = MIN0(Q(J), 108)
57  Q(J) = MAX0(Q(J), 1)

DO 58 J=1,108
58  LINE(J) = 1H

  LINE(Q(1)) = 1H(
  LINE(Q(3)) = 1H(
  LINE(Q(2)) = 1HX
  LDA(NTHERM)    SUB(1)    AJP(59)
  LDA(NFIRST)    AJP(59)

  WRITE(4,121)
59  WRITE(4,122) ID1, (LINE(J), J=1,96), FSUM, FSTDV
122  FORMAT(1H0 1X A8, 2X 96A1, F6.2, *((* F4.2 *)* /)
  SLJ(65)

60  LDA(ICP)      AJP(61)
  LDA(NFIRST)    AJP(61)

  WRITE(5,118) T, FLLO(ICP), FLHI(ICP)
  N5 = 5
  SLJ(800).      IDENTIFY UNITS OF COMPUTER PLOTS
61  WRITE(4,123) T, ID1, ID2
  WRITE(4, 145) NSA, NSF, KS, NA, NB, KP, ANGA, ANGB
  WRITE(4, 144) (ICODE(I), I=1,NC)
  WRITE(4,124) (NGOOD(I),I=1,NC)
124  FORMAT( 12HOGOOD POINTS 17X 13I7 /)
  WRITE(4,125) (SUM(I), I=1,NC)
125  FORMAT(12HMEAN VECTOR 17X 13F7.3)
  SING = 0.

DO 62 I=1,NC
  IF(B(I,I) .LT. 0.) 63, 64

63  SING = 1.
  B(I,I) = 0.
  STDEV(I) = 0.
  SLJ(62)

64  STDEV(I) = SQRT(B(I,I))
62  CONTINUE

  WRITE(4,126) (STDEV(I), I=1,NC)
126  FORMAT(20HSTANDARD DEVIATION 9X 13F7.3)
  WRITE(4,127)
127  FORMAT(19HOCORRELATION MATRIX )

DO 99 I=1,NC
  CALL ZERO(CORR(1), CORR(I))
  CORR(I) = 1.
  LDA1(STDEV)    AJP(67)
  I1 = I-1

DO 66 J=1,I1
  LDA2(STDEV)    AJP(67)
66  CORR(J) = B(J,I)/STDEV(I)/STDEV(J)

```

```

67   LDA3(FLL0)      AJP(97)
      WRITE(4,128) FLL0(I), FLHI(I), (CORR(J), J=1,I)
128  FORMAT(1H0 1X F4.2,4H TO F5.2, 14H MICRON BAND      13F7.3)
      SLJ(99)
97   WRITE(4,146) ICCDE(I), (CORR(J), J=1,I)
146  FORMAT(1H0 24X I2, 2X 13F7.3)
99   CONTINUE

      LDA(NGCOD)      SUB(NC)      AJP2(68).  TEST FOR SINGULAR MATRIX
      WRITE(4, 129) NGOOD(0), NC
129  FORMAT(1H0 I2, * POINTS AND* I3, * CHANNELS, SO SINGULAR *
1    *CCVARIANCE MATRIX *)
      SLJ(69)

68   LDA(SING)      AJP(70)

      WRITE(4,130)
130  FORMAT(/ * SINGULAR MATRIX *)
      SLJ(69)

70   CALL POSDEF(B(0,1), NC, 13, 1, 1H0, SUM(0))

69   WRITE(4,131)
131  FORMAT(1H1 *MEAN VECTOR AND COVARIANCE MATRIX * /)
      WRITE(4,102) (SUM(I), I=1,NC), ((B(I,J), J=1,NC), I=1,NC)
      WRITE(4, 109)
      WRITE(4, 144) (ICODE(J), J = 1, NC)
      WRITE(4, 113) (BOT(I), I=1,NC)
113  FORMAT(1H0 8X*LOWER BOUND EDIT*2X 13F7.3/)
      WRITE(4, 112)
112  FORMAT(1H0 11X*PERCENTILE*/12X*-----*/)

      DO 114 JJ=1,21
      WRITE(4, 114) PCILE(JJ), (ISCN(JJ,J),J=1,NC)
114  FORMAT(11X,F8.1, 8X 13F7.3)

      WRITE(4, 115) (TOP(I), I=1,NC)
115  FORMAT(1H0 8X*UPPER BOUND EDIT*2X 13F7.3/)
      WRITE(4, 105)
      WRITE(4, 123) TITLE, ID1, ID2
      WRITE(4, 132)
132  FORMAT(1H0 33X*SEMILOG PLOT OF MEAN +-1 STANDARD DEVIATION*)
      LDA3(FLL0)      AJP(71)

      WRITE(4,133)
133  FORMAT(1H+ 104X 15HWAVELENGTH BAND / 108X 9H(MICRONS) )
      SLJ(72)

71   WRITE(4,134)
134  FORMAT(1H0 108X 7HCHANNEL )

72   P2 = 0.
      P1 = 100.

      DO 73 I=1,NC
      P2 = AMAX1(SUM(I), P2)
73   P1 = AMIN1(SUM(I), P1)

      P2 = 1.1*P2
      P1 = 0.9*P1

```

```

P2 = AMIN1(100., P2)
P1 = AMAX1(0.01, P1)

FSCALE = ALOG10(P2/P1)
NSCALE = 4./FSCALE
NSCALE = MIN0(4, NSCALE)
NSCALE = MAX0(2, NSCALE)
SCALE = 25*NSCALE
FMID = SQRT(P2*P1)
FSCALE = 50./SCALE
P4 = FMID/(10.**FSCALE)
P4 = AMAX1(P4, 0.02)

ENI4(19)      LDA(P4)
+THS4(A+1)    SLJ(92)

JAC = L+1
IAC = ((18-JAC)*7-1)/6
IAD = MAX0(IAC, -1)
SHIFT = 1. - ALOG10(A(JAD+1))*SCALE

DO 80 I=1,NC
  II = ICODE(I)
  P(2) = SUM(I)
  P(1) = P(2) - STDEV(I)
  P(3) = P(2) + STDEV(I)
  LDA(NFIRST)  AJP(78)

  P(4) = 0.01

DO 74 M=5,13
74  P(M) = P(M-1) + 0.01

DO 75 M=14,22
75  P(M) = P(M-1) + 0.1

DO 76 M=23,31
76  P(M) = P(M-1) + 1.

DO 77 M=32,36
77  P(M) = P(M-1) +10.

DO 81 J=1,36
  LDA2(P)      AJP(79)      AJP3(79)
  Q(J) = ALOG10(P(J))*SCALE + SHIFT
  Q(J) = MIN0(Q(J), 108)
  Q(J) = MAX0(Q(J), 1)
  SLJ(81)
79  Q(J) = 1
81  CONTINUE

DO 82 J=1,108
82  LINE(J) = 1H

DO 83 M=4,36
83  LINE(Q(M)) = 1H.

  LINE(108) = 1H

```

```

DO 36 IPQ=Q(1),Q(3)
36  LINE(IPQ) = 1H-

    LINE(Q(1)) = 1HI
    LINE(Q(3)) = 1HI
    LINE(Q(2)) = 1H*
    LDA3(FLL0)      AJP(84)

    WRITE(4,135) LINE, FLLO(I), FLHI(I)
135  FORMAT(/1H0 108A1, F4.2, 2H - F5.2/)
    LDA(ICP)        AJP(80)      SLJ(85)

84  WRITE(4,136) LINE, II
136  FORMAT(/1H0 108A1, 2X 12 /)

85  LDA(II)          SUB(ICP)      AJP1(80)
    PP(2) = P(2)
    PP(1) = P(1)
    PP(3) = P(3)

DO 86 M=1,3
    QP(M) = PP(M)*12. + 1.
    QP(M) = MIN0(QP(M), 100)
86  QP(M) = MAX0(QP(M), 1)

DO 87 J=1,100
87  MINE(J) = 1H

    LDA(NFIRST)      AJP(88)

    WRITE(5,121)

88  DO 38 IPQ=QP(1),QP(3)
38  MINE(IPQ) = 1H-

    MINE(QP(1)) = 1HI
    MINE(QP(3)) = 1HI
    MINE(QP(2)) = 1H*
    WRITE(5,122) ID1, (MINE(J), J=1,96), SUM(I), STDEV(I)
80  CONTINUE

    MJM = 28/NSCALE + 2
    FMT(1) = 8H1 1X

DO 89 JM=2,MJM
89  FMT(JM) = LABEL(JM+IAD, NSCALE-1)
    FMT(MJM+1) = 8H)

    WRITE(4,FMT)

    N5 = 4
    SLJ4(800).      IDENTIFY UNITS OF COMPUTER PLOTS

    LDA(W)          AJP(65)
    WRITE(W,140) (TITLE(I), I=1,9), ID1, 57B
140  FORMAT( 8A8, A6, 2X, A8, 1X, R1)
    ENCODE(10,102,DATA) (SUM(I),I=1,NC),((B(I,J),J=1,NC),I=1,NC)
    NCARD = (NC + NC*(NC+1)/2 + 4)/5
    J = 0

```



```

DO 155 I=1,NCARD
WRITE(W,141) (DATA(L), L=J,J+9), I, 57B
141  FORMAT(9A8, A3, 3X, I2, 1X, R1)
155  J = J + 10

65   LDA(NSF)          SUB(NSB)      AJP1(156)
161  SLJ4(600).         RE-INITIALIZE FOR NEXT REGION
      SLJ(2).           CALL PROCESS(DATA, 2)

156  NSA = NSF + KS
      LDA(NSB)          SUB(NSF)      SUB(NSK)      AJP2(157)
      NSR = NSB - NSF
      NSF = NSB
      SLJ(158)

157  NSF = NSF + NSK
158  LDA(DELTA)         AJP(160)

      DECODE(1,142,ID1) XID
      XID = XID + DELTA
159  ENCODE(1,142,ID1) XID
      DECODE(1,143,ID1) IH
143  FORMAT(4X, I1, 3X)
      LDA(IH)           SUB(6)        AJP1(160)
      XID = XID + 40.
      SLJ(159)

160  SLJ4(600).         RE-INITIALIZE FOR NEXT REGION
      IF(NSF-NSA) 2,2,45

90   NSB = IS - KS
      NSF = NSB
      NSR = NSF - NSA + 1
      NGCOD = ((NSB-NSA)/KS+1)*((NB-NA)/KP+1)
      LDA(NSR)          SUB(2)
      AJP3(161)         SLJ(52)

200  SLJ(*).           TEST EACH POINT. IF IT PASSES THE TESTS, CUMULATE SUMS
      IF(DATUM(NC1) .EQ. BADPT .AND. DATUM(NC2) .EQ. BADPT) 200, 201
201  SLJ1(203)

DO 210 J=1,NC
IF(DATUM(J) .GT. TOP(J) .OR. DATUM(J) .LT. BOT(J)) 200, 210
210  CONTINUE

203  DO 230 I=1,NC
      LDA1(DATUM)       SUB(BADPT)    AJP(230)
      SUM(I) = SUM(I) + DATUM(I)
      NGCOD(I) = NGCOD(I) + 1

DO 220 J=I,NC
      LDA2(DATUM)       SUB(BADPT)    AJP(220)
      B(I,J) = B(I,J) + DATUM(I)*DATUM(J)
      FN(I,J) = FN(I,J) + 1.
220  CONTINUE

230  CONTINUE

```

```

      SLJ(200)

300  SLJ(*).      GET BOUNDS FOR COLLECTED POINTS, THEN CUMULATE SUMS
      SLJ1(311)

      DO 310 J=1,NC
      CALL MEDIAN(V(0,J), SCAN, IP, CENTER, QDEV)
      ISCN(1,J) = SCAN(1)

      DO 330 JJ=1,20
      NUMB = JJ*(IP-1)/20 + 1
330  ISCN(JJ+1,J) = SCAN(NUMB)

      QDEV = AMAX1(QDEV, 0.867/CONV, EDIT)
      TOP(J) = CENTER + 5.77*QDEV
310  BOT(J) = CENTER - 5.77*QDEV

C      .001 LEGITIMATE PTS. EXCLUDED. VARIANCE DECREASE .LT. 1 0/0

311  DO 320 N=1,IP

      DO 315 J=1,NC
315  DATUM(J) = V(N,J)

      SLJ4(200).      TEST POINTS AND CUMULATE SUMS
320  CONTINUE

      SLJ(300)

700  SLJ(*).      SELECT INPUT MEDIUM
      WRITE(9, 701)
701  FORMAT(* INPUT MEDIUM= *)
      READ(9, 104) REPLY
      LDA(REPLY)      ALS(6)
      STA(R).      R=INPUT MEDIUM
      S = 9
      IF(REPLY .NE. 1HT) 702, 700
702  LIU1(TEST)      IJP1(700)      .IF BATCH, SAVE TYPEWRITER OUTPUT
      S = 23
      SLJ(700)

800  SLJ(*).      IDENTIFY UNITS OF COMPUTER PLOT
      GO TO (153, 152, 151), ITERM

151  WRITE(N5,137)
137  FORMAT(// 1H9 56X *VOLTS* )
      SLJ(800)

152  WRITE(N5,138) ICST
138  FORMAT(/ 1H9 67X, 12 / 1H9 52X *REFLECTANCE X 10 * )
      SLJ(800)

153  WRITE(N5,139) ICST
139  FORMAT(/ 1H9 77X, 12 / 1H9 42X *RADIANCE (WATT/SQ.CM./CM./STER) X
110 * )
      SLJ(800)

92  CALL STOP(7HBAD RUN)
      END $ PAGE

```

```

      FUNCTION  PCSDEF(A, NN, IDA, ISW, CC, MEAN, NTAPE)
C  USES  2*N**2 + N + 3  SPACES IN ERASABLE
      DIMENSION A(IDA,IDA), MEAN(IDA), E(1)
      REAL MEAN
      INTEGER CC, L, N, B, Q, D
      DATA(L=4)
      LOC(ERAS = 63)
      LIN(IJ,JJ) = N*(JJ-1) + II

      N = NN
      IF(N.EQ.1) 25, 2
2     N1 = N + 1
      N2 = N*N + 1
      ENA(E) SUB(ERAS)
      SCM(-OB) STA(B)
      ADD(N1) STA(Q)
      ADD(N2) STA(D)
      ENA(NTAPE)  AJP(1)
      L = NTAPE

1     CALL MXMCV(0, A, E(D), N, N, IDA, N)
      CALL JACOBIN(E(C),N,N,E(B), E(Q), -1)
      WRITE(L,105) CC
105  FORMAT(A1, *EIGENVALUES (YVAR)  YMU      COEFFICIENTS OF THE
      1X'S * /)

      POSDEF = 0.
      DO 24 I = 1,N
      VAR = E(C + LIN(I,I))
      IF(VAR) 5, 5, 6
5     POSDEF = 1.
6     YMU = 0.
      IF(ISW) 10, 15, 10
10    DO 12 J = 1,N
12     YMU = YMU + E(Q + LIN(J,I))*MEAN(IJ)
15    WRITE(L,115) VAR, YMU,(E(Q + LIN(IJ,I)), J=1,N)
115  FORMAT(1H 2E13.5, 2X 13F7.4 / (29X 13F7.4))
24    CONTINUE
25    RETURN
      END

```

```

MACHINE A2F3 (CHAN, SMOOTH, A,B,C,D,E,F,G,H)
LOC (Z=0,POINT=30,ERP=24,MTR=25,RNX=11,RDB=100,WRB=300,CNT=60)
RSV (A1=0)

```

```

*****
*  SIMPLE MESSAGES  *
*****
CON (MES1=2062612363702040B,N1=2045462061216400B). BATCH - NO A/D
CON (MES2=2061026620615167B,N2=2065515146510000B). A2F ARG ERRUR
CON (MES4=2023712343653732B,N3=0000000000000000B). TITLE
CON (MES6=2020204461453020B) . MANY
CON (MESCORE=2223465161676520B,M2=4371447123000000B).STORAGE LIMITA
CON (WONKITY=2026464542712330B,M3=1452733220462423B).OUTPUT TAPE
CON (M4 =4724232023614765B,M5=2045462320516561B).NOT READY
CON (M6 =6430730000000000B)
CON (DONE =6446456520267123B,M7=7020236147653700B).DONE WITH TAP
CON (WARN =2046632361432061B,M8=7167227300000000B).OCTAL ARGS.
*****
*  TAPE SELECT CODES  *
*****

```



```
CON (T0=0) .DUMMY
CON (T1=0140000000042011B,T2=0240000000042021B) .1 2
CON (T3=0340000000042031B,T4=0440000000042041B) .3 4
CON (T5=0540000000043011B,T6=0640000000043021B) .5 6
CON (T7=0740000000043031B,T8=1040000000043041B) .7 8

*****
* MASKS *
*****

CON (MASK11=0 777 000 777 000 777 B)
CON (MASK12=0 377 000 377 000 377 B)
CON (MASK21=0 000 777 000 777 000 B)
CON (MASK22=0 000 377 000 377 000 B)
CON (MASK3= 4 003 777 777 777 777 B)
CON (MASK0= 4 003 774 003 774 003 B)
CON (MASKE= 7 774 003 774 003 777 B)
CON (MASK =0 400 400 400 400 400 B)

CON (MASKP1= 000 000 000 000 777 0B)
CON (MASKP2= 000 000 000 777 000 0B)
CON (MASKP3= 000 000 777 000 000 0B)
CON (MASKP4= 000 777 000 000 000 0B)
CON (MASKP5= 777 000 000 000 000 0B)

CON (MASK1=0, MASK2=0)

*****
* JOB STATISTICS MESSAGES *
*****

LOC (NUMSTIX=14)
CON (ERRORS=0, SHORT=0, BURNED=0, LOST=0, LONG=0, STRETCH=0)
CON (SCANS=0, LINES=0, SIZE=0, NSS=0, NCHAN=0, SMOOTH=0)

CON (JJ=0, J1=2361476520655151B, J2=4651220000000000B).TAPE ERRORS
CON (J3=0, J4=2020234646202270B, J5=4651230000000000B). TOO SHORT
CON (J6=0, J7=2020612320222361B, J8=5123000000000000B). AT START
CON (J9=0, J0=2043462223000000B, JA=0000000000000000B). LOST
CON (J8=0, JC=2023517144446564B, JD=0000000000000000B). TRIMMED
CON (JE=0, JF=2022235165236370B, JG=6564000000000000B). STRETCHED
CON (JH=0, JI=2263614522000000B, JK=0000000000000000B). SCANS
CON (JL=0, JM=4371456522000000B, JN=0000000000000000B). LINES
CON (JO=0, JP=2646516422214371B, JQ=4565000000000000B). WORDS/LINE
CON (JR=0, JS=2261444743652221B, JT=4371456500000000B). SAMPLES/LINE
CON (JU=0, JV=6370614545654320B, JW=6461236100000000B). CHANNEL DATA
CON (JX=0, JY=2346200120224446B, JZ=4623700000000000B). TO 1 SMOOTH

*****
* OTHER JUNK *
*****

CON (ENDREEL=1765405165654317B)
CON (SPACES =20202020202020B)
CON (FIRST =00000017777777B)
CON (SMOOTH =SMOOTHED)

*****
* BEGIN MAIN PROGRAM *
*****

1ENT SLJ (*) SLJ (L+11) .EXIT/ENTRY
OARG BSS {1} .CHANNELS
```

9ARG	BSS (1)		.SMOOTHING FACTOR	A2F30860
1ARG	BSS (8)		.TAPES	A2F30870
	LIL1(POINT)	ENA (MES1)	.TEST FOR NEW RESIDENT	A2F30880
	IJP1(L+1)	SLJ (L+4)	.GO AHEAD IF OLD RESIDENT	A2F30890
	LDQ1(Z+2)	STQ (L+2)	.	A2F30900
	ENI1(ERP)	SIL1(L+1)	.	A2F30910
	BSS (1)		.EXIT IF IN BATCH	A2F30920
	ENA (8)	ENI1(8)	.OK. DO ARGUMENTS	A2F30930
	THS1(1ARG)	SLJ (L+4)	.ERROR IF ARGUMENT LARGER THAN 8	A2F30940
1AER	ENA (0)	ENI1(7)	.ARGUMENT ERROR	A2F30950
	STA1(1ARG)	IJP1(L)	.CLEAR FOR NEXT CALL	A2F30960
	ENA (MES2)	SLJ4(ERP)	.TYPE MESSAGE/EXIT	A2F30970
	ENA (0)	ENI1(8)	.GET NUMBER OF ARGUMENTS	A2F30980
	THS1(1ARG)	SLJ (1AER)	.	A2F30990
	SIU1(2EOT)	ENI2(TO)	.SAVE COUNT, GET ADDRESS OF TABLE	A2F31000
2ARG	LDA1(1ARG)	SAL (L+2)	.GET ARGUMENT	A2F31010
	AJP (1AER)	AJP3(1AER)	.ERROR IF ZERO OR NEGATIVE	A2F31020
	SIL1(L+1)	LDQ2(*)	.SET SELECT CODE	A2F31030
	STQ1(9SEL)	ENI3(*)	.SAVE COUNT	A2F31040
	EQS3(1ARG)	SLJ (L+2)	.SEARCH FOR DUPLICATE ARGUMENTS	A2F31050
	SLJ (1AER)		.ERROR IF DUPLICATE ARG FOUND	A2F31060
	ENA (0)	STA1(1ARG)	.CLEAR ARGUMENT	A2F31070
	IJP1(2ARG)	ENI1(NUMST(X-1)	.LOOP FOR MORE TAPES	A2F31080
	STA1(ERRORS)	IJP1(L)	.CLEAR JOB STATISTICS	A2F31090
	LDA (OARG)	AJP (1AER)	.ERROR IF NCHAN .EQ. 0	A2F31100
	THS (16)	SLJ (1AER)	.ERROR IF NCHAN .GT. 16	A2F31110
	AJP3(1AER)	STA (NCHAN)	.IF NCHAN .GT.-0, STORE IT	A2F31120
	SAL (4IA)	ALS (1)	.	A2F31130
	SAU (6IA)	INA (1)	.	A2F31140
	SAU (2TST)	SCM (-OB)	.	A2F31150
	SAU (2IA)	LAC (NCHAN)	.	A2F31160
	SAL (1IA)	SAU (3IA)	.	A2F31170
	ENA (A1+1)	ADD (NCHAN)	.	A2F31180
	SCM (-OB)	SAU (1AGN)	.	A2F31190
	LDA (9ARG)	STA (SMOOTH)	.STORE SMOOTH FACTOR	A2F31200
	THS (16)	SLJ (1AER)	.ERROR IF SMOOTH .GT. 16	A2F31210
	INA (-3)	AJP3(1AER)	.ERROR IF SMOOTH .LE. 2	A2F31220
	INA (2)	SAU (101E)	.SET UP SMOOTH LOOP LIMIT	A2F31230
	ENI1(2)	ENA (7)	.	A2F31240
	THS1(OARG)	SLJ (L+2)	.TEST FOR SMOOTH OR CHAN OVER 7	A2F31250
	ENA (WARN)	SLJ4(MTR)	.ISSUE WARNING MESSAGE	A2F31260
	ENI1(0)	ENQ (1)	.	A2F31270
	ENA (0)	DVI (SMOOTH)	.	A2F31280
	SCA6(2)	SCL (MASK3)	.	A2F31290
	IJP6(L+1)	ENI1(1)	.	A2F31300
	IJP6(L+1)	ENI6(1)	.	A2F31310
	SIU1(100S)	SIU1(101S)	.	A2F31320
	LDQ1(MASK11)	STQ (MASK1)	.	A2F31330
	LDQ1(MASK21)	STQ (MASK2)	.	A2F31340
	SIL6(200)	SIL6(200M)	.	A2F31350
	STA (IKR)	ENI6(A1)	.	A2F31360
	ENQ6(-17416B)	ENA (MESCORE)	.	A2F31370
	QJP6(ERP)	ENA6(1200)	.CRASH IF TOO NEAR END OF CORE	A2F31380
	SAL (Z+5)	SAL (OSTX)	.STORE END OF INITIAL READ BUFF	A2F31390
	SCM (-OB)	SAL (2K)	.	A2F31400
	LDA (Z+7)	SAU (9EOT)	.	A2F31410
	SAU (1OUT)	ENQ (37777B)	.SAVE INTERRUPT ROUTINE ADDRESS	A2F31420
	ENA (1ENT)	LIL1(CNT+5)	.	A2F31430
	MEQ7(CNT+7)	SLJ (L+3)	.FIND A2F IN TABLE	A2F31440
	INI1(1)	SIL1(CNT+5)	.REMOVE REMAINING ENTRIES	A2F31450
	ENA (A1)	SAL (CNT+3)	.SET END OF A2F	A2F31460
	SIL (1FLG)	SIU (8NXT)	.STORE ZERO	A2F31470
	SIL (1EOT)	ENA (37777B)	.	A2F31480
	SAL (7NXT)	SAU (9NXT)	.STORE 37777	A2F31490

SAL (2FLG)	SAL (OFLG)	.STORE NON-ZERO	A2F31500
ENA (WRB)	SAU (1WRT)	.MORE ADDRESSES	A2F31510
ENA (WRB+54)	SAU (OWRT)	.	A2F31520
SAL (Z+4)	ENI1(A1)	.	A2F31530
ENQ (0)	STQ1(Z)	.CLEAR CORE	A2F31540
ISK1(37777B)	SLJ (L-1)	.	A2F31550
ENA (2WRT)	SAU (OTAP)	.	A2F31560
ENA (OEND)	SAL (1TYP)	.	A2F31570
ENA (OIN)	SAU (OA2D)	.	A2F31580
EXF (101B)	EXF (2000B)	.CLEAR FAULT INT., STOP CLOCK	A2F31590
			A2F31600
			A2F31610
*****			A2F31620
* FIRST TAPE RECORD *			A2F31630
*****			A2F31640
ENA (MES4)	SLJ4(MTR)	.TYPE *TITLE*	A2F31650
ENA (9)	SLJ4(RNX)	.	A2F31660
ENI1(32)	ENA (0)	.	A2F31670
STA (WRB+21)	IJP1(L)	.CLEAR LAST OF BUFFER	A2F31680
STA (WRB+18)		.	A2F31690
ENI3(11)	ENI2(95)	.*LOOP	A2F31700
ENI1(7)		.*TO	A2F31710
LDA2(RDB)	LRS (6)	.*PACK	A2F31720
INI2(-1)	IJP1(L-1)	.*THE	A2F31730
STQ3(WRB+1)	IJP3(L-3)	.*TITLE	A2F31740
LDA (NSS)	STA (WRB+13)	.	A2F31750
LDA (1.)	STA (WRB+14)	.	A2F31760
LDA (1.)	STA (WRB+15)	.	A2F31770
LDA (NCHAN)	STA (WRB+16)	.	A2F31780
LDA (51.2)	STA (WRB+17)	.	A2F31790
ENA (1)	STA (WRB+19)	.	A2F31800
LDA (SMOHD)	STA (WRB+20)	.	A2F31810
LDA (FIRST)	STA (WRB)	.	A2F31820
ENI6(0)	SLJ4(OSEL)	.SELECT A TAPE	A2F31830
LDA (OTAP)	SAU (L+1)	.	A2F31840
EXF (*)	EXF (11100B)	.INTERRUPT ON END OF WRITE	A2F31850
ENA (T1)	SAL (Z+1)	.	A2F31860
EXF1(T0)	EXF (50341B)	.START TYPEWRITER AND CONVERTER	A2F31870
EXF5(A1)	ENI4(0)	.START CHANNEL 5	A2F31880
			A2F31890
*****			A2F31900
* GET FIRST SCAN *			A2F31910
*****			A2F31920
EXF7(10B)	SLJ (OSTX)	.EXIT IF TYPEWRITER DONE	A2F31930
EXF7(50324B)	SLJ (L-1)	.	A2F31940
LIU6(Z+5)	EXF (50341B)	.GET ADDRESS, START CONVERTER	A2F31950
EXF5(A1)		.RESTART CHANNEL	A2F31960
OAGN RAO (SCANS)	STA (BURNED)	.COUNT SCAN	A2F31970
STA (LOST)		.	A2F31980
IAGN ENA6(*)	AJP3(L-5)	.REJECT SCAN IF TOO SHORT	A2F31990
			A2F32000
*****			A2F32010
* INITIALIZE ADDRESSES *			A2F32020
*****			A2F32030
11A SIU6(3TST)	ENA6(*)	.	A2F32040
SCM (-OB)	SAU (1IN5)	.TEST A1 UNDERSIZE	A2F32050
21A INA (*)	SAU (1TST)	.TEST A1 OVERSIZE	A2F32060
INA (A1+1)	SAU (2IN6)	.DISPLACEMENT FOR STRETCHING A2	A2F32070
SCM (-OB)	SAU (1IN6)	.DISPLACEMENT FOR STRETCHING A1	A2F32080
31A INA (*)	MUI (5)	.	A2F32090
DVI (NCHAN)	STA (WRB+13)	.NUMBER OF SAMPLES	A2F32100
			A2F32110
			A2F32120

	STA (NSS)	MUI (NCHAN)	.	A2F32130
	CVI (5)		.	A2F32140
	GJP (L+1)	INA (1)	.NUMBER OF WORDS PER LINE	A2F32150
	STA (SIZE)	SAL (5NXT)	.	A2F32160
	SAL (1NXT)	INA (-1)	.	A2F32170
	SAU (1INX)	SAL (1INX)	.	A2F32180
	SAL (5IA)	INA (A1)	.	A2F32190
	SAU (1IN7)	INA (1)	.LIMIT FOR STRETCHING A1	A2F32200
	SCM (-0B)	SAU (1IN4)	.	A2F32210
4IA	SAL (1K)	ENA6(*)	.TEST FOR STRETCHING NEEDED IN A1	A2F32220
	SAU (2FLG)	SAU (1IN1)	.START OF A2	A2F32230
	SAL (1IN3)	SAL (2IN2)	.	A2F32240
5IA	SAU (1K)	INA (*)	.	A2F32250
	SAU (2IN7)	INA (1)	.LIMIT FOR STRETCHING A2	A2F32260
	SCM (-0B)	SAU (2IN4)	.TEST FOR STRETCHING NEEDED IN A2	A2F32270
	ENA6(0)	INA6(0)	.	A2F32280
	SCM (-0B)	INA (A1)	.	A2F32290
	SAU (2IN5)	SCM (-0B)	.TEST FOR A2 UNDERSIZE	A2F32300
6IA	INA (*)	SAU (100M)	.START OF M	A2F32310
	SAU (101M)	SAU (200M)	.	A2F32320
	SAU (2IN1)	SAL (2IN3)	.	A2F32330
	SAU (0K)	ADD (SIZE)	.	A2F32340
	SAU (100N)	SAU (101N)	.START OF N	A2F32350
	SAU (200)	ADD (SIZE)	.	A2F32360
	SAU (7WRT)	ADD (SIZE)	.START OF TAPE CORRECTION AREA	A2F32370
	SAU (6NXT)	SAU (0WRT)	.START OF FIRST OUTPUT BUFFER	A2F32380
	SAL (6WRT)	SAU (2NXT)	.	A2F32390
	INA (-1)	SAL (3K)	.LAST WORD OF TAPE CORRECTION AREA	A2F32400
	SAU (0NXT)		.	A2F32410
				A2F32420
				A2F32430
				A2F32440
				A2F32450
				A2F32460
	EXF7(10B)	SLJ (0STX)	.EXIT IF TYPEWRITER DONE	A2F32470
	EXF7(50324B)	SLJ (L-1)	.WAIT FOR END OF LINE	A2F32480
	LIU6(Z+5)	EXF (50341B)	.SAVE SIZE, RESTART CONVERTER	A2F32490
1TST	ENA6(*)	AJP2(2BIG)	.JUMP IF OVERSIZE	A2F32500
2TST	INA (*)	AJP2(0K)	.JUMP IF SIZE IS OK	A2F32510
	EXF5(A1)	SIU6(L+3)	.TOO SMALL, START OVER, SAVE SIZE	A2F32520
	ENA (0)	STA6(Z)	.CLEAR LEFTOVER DATA	A2F32530
3TST	ISK6(*)	SLJ (L-1)	.LOOP (CLEAR AN EXTRA WORD)	A2F32540
	ENI6(*)	SLJ (0AGN)	.RESTORE SIZE, GO RE-INITIALIZE	A2F32550
2BIG	EXF5(A1)	SLJ (0AGN)	.TOO BIG, GO RE-INITIALIZE	A2F32560
				A2F32570
				A2F32580
				A2F32590
				A2F32600
				A2F32610
OK	ENI5(*)	SIL5(Z+5)	.START BUFFER FOR NEXT SCAN	A2F32620
1K	EXF5(*)	ENA6(*)	.	A2F32630
	AJP2(L+1)	RAO (STRETCH)	.IF SCAN WAS SHORT, COUNT STRETCH	A2F32640
2K	RAO (SCANS)	ENA6(*)	.COUNT SCAN	A2F32650
	AJP3(L+1)	RAO (LONG)	.IF SCAN WAS LONG, COUNT A TRIM	A2F32660
3K	ENA (-0)	STA (*)	.STORE THE TAPE CORRECTION WORD	A2F32670
	ENA (0INT)	EXF4(WRB)	.START THE TAPE	A2F32680
	EXF (50342B)	EXF (10B)	.SELECT CONVERTER AND CONSOLE INTS	A2F32690
	SAL (Z+7B)	SLJ (1FLG)	.CONNECT THE INT. ROUTINE AND GO	A2F32700
				A2F32710

 * WAIT FOR GOOD SCAN *

 * GOOD SCAN, START EVERYTHING *



```
*****
*
*   PROCESS THE DATA (FROM ADTEST2)
*   (INDEXED CARDS ARE UNMODIFIED)
*   (MAY NOT USE INDEX 6)
*   (SMOOTHES OF 3 THROUGH 16)
*
*****
A2F32720
A2F32730
A2F32740
A2F32750
A2F32760
A2F32770
A2F32780
A2F32790
A2F32800
A2F32810
A2F32820
A2F32830
A2F32840
A2F32850
A2F32860
A2F32870
A2F32880
A2F32890
A2F32900
A2F32910
A2F32920
A2F32930
A2F32940
A2F32950
A2F32960
A2F32970
A2F32980
A2F32990
A2F33000
A2F33010
A2F33020
A2F33030
A2F33040
A2F33050
A2F33060
A2F33070
A2F33080
A2F33090
A2F33100
A2F33110
A2F33120
A2F33130
A2F33140
A2F33150
A2F33160
A2F33170
A2F33180
A2F33190
A2F33200
A2F33210
A2F33220
A2F33230
A2F33240
A2F33250
A2F33260
A2F33270
A2F33280
A2F33290
A2F33300
A2F33310
A2F33320
A2F33330
A2F33340
A2F33350

*****
*   GET INPUT BUFFER TO EMPTY
*
*****

100  ISK7(0FLG)      SLJ (L+3)      .WHICH BUFFER
1FLG  ENA (A1)       ISK (*)        .WAIT FOR BUFFER 1
      SAL (0FLG)     SLJ (L+3)      .
2FLG  ENA (*)        ISK (*)        .WAIT FOR BUFFER 2
0FLG  SIL (0FLG)     ENI (*)        .
      EXF7(10B)      SLJ (00UT)    .EXIT IF DONE
11NX  ENI1(*)        ENI2(*)        .PLACE BUF SIZE IN INDEX REGISTERS

*****
*   FIRST PASS - STORE VALUES
*
*****

      IJP4(100C)      SAU(L+1)      .SET UP
      LDA2(*)         SCM(MASK)     .BIAS DATA UPWARD BY 5.0 VOLTS
100S  ARS (*)         LDQ (MASK2)   .SHIFT RIGHT FOR MAXIMUM CAPACITY
100M  STL2(*)         LDQ (MASK1)   .STORE EVEN COMPONENTS
100N  STL2(*)         IJP2(L-3)     .STORE ODD COMPONENTS
      SLJ (100D)      .CONTINUE

*****
*   REMAINING PASSES - ADD TO STORAGE
*
*****

100C  IN14(1)        SAU(L+1)      .SET UP
      LDA2(*)         SCM(MASK)     .BIAS DATA UPWARD BY 5.0 VOLTS
101S  ARS (*)         STA (DATUM)   .SHIFT RIGHT FOR MAXIMUM CAPACITY
      LDQ (DATUM)     LDL (MASK2)   .BRING EVEN VALUES INTO ACCUMULATOR
101M  RAD2(*)         LDL (MASK1)   .ODD VALUES INTO ACCUMULATOR
101N  RAD2(*)         IJP2(L-4)     .CONTINUE

*****
*   RELEASE THE INPUT BUFFER
*
*****

100D  ISK7(0FLG)      SLJ (L+2)      .WHICH BUFFER
      SIL1(2FLG)     SLJ (L+2)      .RELEASE BUFFER 2
      SIL1(1FLG)     ARS (0)        .RELEASE BUFFER 1
101E  ISK4(*)        SLJ (100)      .CONTINUE

*****
*   GET EMPTY OUTPUT BUFFER
*
*****

5NXT  ENA7(7NXT)      INA (1)      .
      SAU (200S)      INA (*)      .STORE START OF NEW BUFFER
      SAL (7NXT)      SAL (L+3)    .STORE END OF NEW BUFFER
      INA (-37777H)   AJP3(L+2)    .DOES NEW BUFFER EXIST
6NXT  ENA (*)        SLJ (L-3)      .NO-GET FIRST ONE
      -              ISK7(*)        .WAIT FOR BUFFER EMPTY
```


19

```

*****
* BUFFER 1 *
*****
1IN1  ENA (*)          SAL (Z+5)      .SET ADDRESS OF END OF BUFFER
      ISK7(1FLG)       SLJ (L+4)      .IS THE BUFFER EMPTY
      EXF (50341B)     SLJ4(0A2D)     ..FULL-START AND GO WAIT
1INQ  RAO (LOST)       RAO (SCANS)    ..COUNT LOST SCAN
      SLJ (L-3)        ..GET NEXT SCAN
      EXF (50341B)     EXF5(A1)      .EMPTY-START CONVERTER AND CHANNEL
      SLJ4(0A2D)       .GO WAIT FOR END OF LINE
      EXF7(51B)        SLJ (L+2)     .TEST FOR CHANNEL ACTIVE
      RAO (LONG)       SLJ (1IN)     .NOT ACTIVE-LONG SCAN-ACCEPT IT
1IN3  LIU6(Z+5)       EXF5(*)        .SAVE ADDRESS-STOP CHANNEL
1IN4  ENA6(*)          AJP2(1IN)     .TEST FOR FULL SCAN
1IN5  ENA6(*)          AJP2(L+2)     .SHORT...STRETCH OR SCRATCH
      RAO (SHORT)     SLJ (1INQ)    ..SCRATCH IT
1IN6  LDA6(*)         STA6(Z)       ..STRETCH IT
1IN7  ISK6(*)         SLJ (L-1)     ..CONTINUE
      RAO (STRETCH)   ARS (O)      ..
1IN   RAO (SCANS)     SIL (1FLG)    .ACCEPT THE SCAN-RELEASE THE BUFF

*****
* BUFFER 2 *
*****
2IN1  ENA (*)          SAL (Z+5)      .SET ADDRESS OF END OF BUFFER
      ISK7(2FLG)       SLJ (L+4)      .IS THE BUFFER EMPTY
      EXF (50341B)     SLJ4(0A2D)     ..FULL-START AND GO WAIT
2INQ  RAO (LOST)       RAO (SCANS)    ..COUNT LOST SCAN
      SLJ (L-3)        ..GET NEXT SCAN
2IN2  EXF (50341B)     EXF5(*)      .EMPTY-START CONVERTER AND CHANNEL
      SLJ4(0A2D)       .GO WAIT FOR END OF LINE
      EXF7(51B)        SLJ (L+2)     .TEST FOR CHANNEL ACTIVE
      RAO (LONG)       SLJ (2IN)     .NOT ACTIVE-LONG SCAN-ACCEPT IT
2IN3  LIU6(Z+5)       EXF5(*)        .SAVE ADDRESS-STOP CHANNEL
2IN4  ENA6(*)          AJP2(2IN)     .TEST FOR FULL SCAN
2IN5  ENA6(*)          AJP2(L+2)     .SHORT...STRETCH OR SCRATCH
      RAO (SHORT)     SLJ (2INQ)    ..SCRATCH IT
2IN6  LDA6(*)         STA6(Z)       ..STRETCH IT
2IN7  ISK6(*)         SLJ (L-1)     ..CONTINUE
      RAO (STRETCH)   SLJ (2IN)    ..

*****
*
* TAPE WRITING LOGIC *
*
*****

*****
* GET A BUFFER OF OUTPUT *
*****
ONXT  ENA (*)          INA (1)
1NXT  SAU (1WRT)       INA (*)
      SAU (9NXT)       SAU (0WRT)
      SAU (0NXT)       SAU (4NXT)
      INA (-37777B)    AJP3(3NXT)
2NXT  ENA (*)          SLJ (1NXT)
3NXT  ENA (20B)        SIU (8NXT)
4NXT  ISK7(*)          SLJ (0WRT)
      SAU (8NXT)       SLJ4(0TAP)
      SLJ (3NXT)

      .
      .STORE START OF NEXT BUFFER
      .STORE END OF NEXT BUFFER
      .
      .DOES NEW BUFFER EXIST
      .NO, CHANGE TO FIRST ONE
      .CLEAR FORCED INTERRUPT
      .IF NEW BUFFER FULL, GO TO WRITE
      .WAIT FOR BUFFER FULL INTERRUPT
      .GO TEST IT AGAIN

```



```
*****
* WRITE A RECORD *
*****
0WRT  ENA (*)          SAL (Z+4)          .SET END
1WRT  EXF4(*)          SLJ4(OTAP)         .START WRITE, WAIT FOR DONE
2WRT  EXF7(*)          SLJ (5WRT)        .TEST FOR PARITY ERROR
3WRT  EXF7(*)          SLJ (5WRT)        .TEST FOR LENGTH ERROR
4WRT  EXF7(*)          SLJ4(OEOT)        .CHANGE TAPE IF END OF REEL
9NXT  SIL (*)          SLJ (ONXT)        .RELEASE BUFFER, GO GET NEXT ONE
*****
* CORRECT TAPE ERRORS *
*****
5WRT  EXF7(*)          SLJ (9WRT)        .TEST FOR END OF TAPE
6WRT  EXF (*)          ENA (*)          .SELECT BCD, SET END OF CLEAR BUFF
        SAL (Z+4)      SLJ4(OBKS)        .BACKSPACE
7WRT  EXF4(*)          SLJ4(OTAP)        .WRITE BLANK TAPE
8WRT  EXF7(*)          SLJ (9WRT)        .TEST FOR END OF TAPE
8WRT1 EXF (*)          SLJ4(OBKS)        .NOT END, BIN MODE AND BACKSPACE
        RAG (ERRCRS)    SLJ (OWRT)        .COUNT ERROR, GO RE-WRITE
9WRT  SLJ4(OBKS)        .BACKSPACE
        SLJ4(OEOT)      .CHANGE TAPE
        RAG (ERRCRS)    SLJ (OWRT)        .COUNT ERROR, GO RE-WRITE
*****
* TAPE CONTROL SUBROUTINES *
*****
*****
* BACKSPACE ONE RECORD *
*****
1BK5  EXF (*)          SLJ4(OTAP)        .BACKSPACE/WAIT READY
OBKS  SLJ (*)          SLJ (L-1)         .EXIT/ENTRY
*****
* END OF TAPE - UNLOAD TAPE AND GET A NEW ONE *
*****
OEOT  SLJ (*)          EXF (*)          .EXIT/ENTRY, SELECT BCD
        ENA (SPACES)    SAL (Z+4)        .SET END OF EOT RECORD
        EXF4(ENDREEL)  SLJ4(OTAP)        .WRITE ENDTAPE RECORD/WAIT READY
1EOT  EXF (*)          ENI6(*)          .UNLOAD TAPE, PUT COUNTER IN INDEX
2EOT  ISK6(*)          ARS (0)          .CYCLE TAPE COUNTER
3EOT  SIL6(1EOT)       SLJ4(OSEL)        .STORE COUNTER, SELECT NEW UNIT
        SLJ (OEOT)      .RETURN
*****
* SELECT TAPES *
*****
9SEL  BSS (8)          .TAPE SELECT CODE TABLE
OSEL  SLJ (*)          LDA6(9SEL)        .EXIT/ENTRY, GET SELECT CODE
        SAU (L+1)      SCL (778)        .STORE XCOU1
        EXF (*)          SAU (1TAP)      .STORE XC000, SELECT TAPE
        INA (1)         SAU (5SEL)      .STORE XC001
        SAU (8WRT1)     INA (1)         .
        SAL (OEOT)      SAU (6WRT)      .STORE XC002
```

A2F34640
A2F34650
A2F34660
A2F34670
A2F34680
A2F34690
A2F34700
A2F34710
A2F34720
A2F34730
A2F34740
A2F34750
A2F34760
A2F34770
A2F34780
A2F34790
A2F34800
A2F34810
A2F34820
A2F34830
A2F34840
A2F34850
A2F34860
A2F34870
A2F34880
A2F34890
A2F34900
A2F34910
A2F34920
A2F34930
A2F34940
A2F34950
A2F34960
A2F34970
A2F34980
A2F34990
A2F35000
A2F35010
A2F35020
A2F35030
A2F35040
A2F35050
A2F35060
A2F35070
A2F35080
A2F35090
A2F35100
A2F35110
A2F35120
A2F35130
A2F35140
A2F35150
A2F35160
A2F35170
A2F35180
A2F35190
A2F35200
A2F35210
A2F35220
A2F35230
A2F35240
A2F35250
A2F35260
A2F35270

INA (1)	SAU (2WRT)	.STORE XC003	A2F35280
INA (1)	SAL (OTAP)	.STORE XC004	A2F35290
INA (1)	SAU (3WRT)	.STORE XC005	A2F35300
INA (1)	SAU (1BKS)	.STORE XC006	A2F35310
INA (1)	SAU (1EOT)	.STORE XC007	A2F35320
SAU (4WRT)	SAU (5WRT)	.	A2F35330
SAU (8WRT)	ARS (0)	.	A2F35340
5SEL EXF7(*)	SLJ (0SEL)	.RETURN IF TAPE READY	A2F35350
*****			A2F35360
* END THE RUN *			A2F35370
*****			A2F35380
			A2F35390
			A2F35400
			A2F35410
			A2F35420
			A2F35430
*****			A2F35440
* NEW TAPE NOT READY, CRASH THE RUN *			A2F35450
*****			A2F35460
EXF (118)	EXF (500008)	.CLEAR INTERRUPT AND CONVERTER	A2F35470
EXF7(118)	EXF1(T1)	.	A2F35480
LDA (Z+5)	SAU (L+1)	.STOP CHANNEL 5	A2F35490
EXF5(*)	ENA (L+2)	.	A2F35500
SAU (Z+7)	SLJ (Z+7)	.LEAVE INTERRUPT MODE	A2F35510
ENI(0)	ENI(0)	.DO NOTHING TO CATCH JUMP	A2F35520
ENA (WONKITY)	SLJ4(MTR)	.WRITE *NO TAPES* MESSAGE	A2F35530
9ECT ENA (*)	SAL (Z+7)	.	A2F35540
SLJ4(0Z)		.PRINT STATISTICS	A2F35550
SLJ (1ENT)		.EXIT	A2F35560
*****			A2F35570
* AFTER INCOMPLETE START *			A2F35580
*****			A2F35590
OSTX EXF5(*)	EXF (500008)	.CLEAR CHANNEL 5	A2F35600
SLJ4(0Z)		.PRINT STATISTICS	A2F35610
SLJ (1ENT)		.EXIT	A2F35620
*****			A2F35630
* AFTER NORMAL RUN *			A2F35640
*****			A2F35650
0OUT LIU1(8NXT)	IJP1(L+2)	.WAIT FOR LAST TAPE WRITE	A2F35660
SLJ (L-1)		.	A2F35670
LDA (OTAP)	SCL (778)	.	A2F35680
SAU (L+1)	SAL (L+1)	.	A2F35690
EXF (*)	EXF7(*)	.CLEAR TAPE INTERRUPT SELECT	A2F35700
INA (3)	SAU (2OUT)	.STORE EOF SELECT CODE	A2F35710
INA (4)	SAL (3OUT)	.STORE UNLOAD SELECT CODES	A2F35720
1OUT ENA (*)	SAL (Z+7)	.RESTORE INTERRUPT	A2F35730
2OUT EXF (*)	SLJ4(0Z)	.WRITE EOF, GO PRINT STATISTICS	A2F35740
ENA (DONE)	SLJ4(MTR)	.ASK IF USER WANTS TAPE UNLOADED	A2F35750
ENA (9)	SLJ4(RNX)	.GO GET REPLY	A2F35760
LDA (RDB)	INA (-308)	.	A2F35770
3OUT AJP1(1ENT)	EXF (*)	.UNLOAD TAPE IF YES	A2F35780
SLJ (1ENT)		.	A2F35790
*****			A2F35800
* PRINT STATISTICS *			A2F35810
*****			A2F35820
0Z SLJ (*)		.EXIT/ENTRY	A2F35830
ENI5(NUMSTIX)	ENA (0)	.	A2F35840
		.	A2F35850
		.	A2F35860
		.	A2F35870
		.	A2F35880
		.	A2F35890
		.	A2F35900
		.	A2F35910

1Z	THS5(ERRORS)	SLJ (0Z)	.SEARCH/EXIT	A2F35920
	ENA5(JJ)	INA5(0)	.GENERATE ADDRESS	A2F35930
	INA5(0)	LDQ5(ERRORS)	.GET VARIABLE	A2F35940
	SAL (3Z)	SAL (4Z)	.SAVE ADDRESS	A2F35950
	QJP3(5Z)	LDA (SPACES)	.JUMP IF MINUS, GET BLANKS IF NOT	A2F35960
	STA7(4Z)	ENI2(6)	.SET BLANKS	A2F35970
2Z	LDA5(ERRORS)	ENQ (0)	.	A2F35980
	DVI (10)	STA5(ERRORS)	.GET LEAST SIGNIFICANT DIGIT	A2F35990
	QJP1(L+1)	ENQ (12B)	.SET TO 12B IF ZERO	A2F36000
3Z	LDL (77B)	LDQ (*)	.PACK DIGIT	A2F36010
4Z	LRS (6)	STQ (*)	.	A2F36020
	LDA5(ERRORS)	AJP (L+2)	.EXIT IF NUMBER DONE	A2F36030
	IJP2(2Z)	SLJ (5Z)	.LOOP/SET MANY IF OVERFLOW	A2F36040
	LDA (SPACES)	LLS (6)	.	A2F36050
	LRS (6)	IJP2(L)	.FILL WITH LEADING BLANKS	A2F36060
	STQ7(4Z)	SLJ (L+2)	.STORE AND GO PRINT	A2F36070
5Z	LDA (MES6)	STA7(4Z)	.SET / MANY /	A2F36080
	ENA7(4Z)	SLJ4(MTR)	.PRINT LINE	A2F36090
	SLJ (1Z)		.LOOP FOR MORE	A2F36100
				A2F36110
				A2F36120

END

```

PROGRAM CSD1(IARG1,IARG2,IARG3)
C  NEEDS PROCESS, UNPACK3, BUFPACK.
C  INPUT ON 3, OUTPUT ON 4.
C  PROGRAM HAS 'CONTINUE FILE' OPTION
C  'TIME' CHANNEL SHOULD BE LAST CHANNEL ON TAPE.

C  DESIGNED PRIMARILY FOR DOING DYNAMIC CLAMPING TO THE DARK LEVEL,
C  AND DYNAMIC SCALING TO THE SUN SENSOR.
C  SCALE FACTORS ARE COMPUTED ONLY WHEN AIRCRAFT ROLL (AS INDICATED
C  BY THE SIGNAL IN A DIGITIZED 'TIME' CHANNEL) IS WITHIN CERTAIN
C  LIMITS. LIMITS WILL BE COMPUTED BY THE PROGRAM.
C  HAS 'DESKEWING' CAPABILITY ('SKEWS =' 0 FOR NO DESKEWING)
C  THE 'CLAMPING REGION' WILL NOT BE DESKEWED PRIOR TO CLAMPING DATA
C  'SCALING REGION' IS DESKEWED PRIOR TO SCALING (MINIMUM SKEW .EQ. 0)
C  -----
C  SAMPLE REGION LINE NO.= 0 TO SKIP COMPUTATION OF LIMITS
C  A SCALE FACTOR OF ' 0.' WILL SIGNAL THAT PREVIOUS VALUE SHOULD BE
C  USED (' 1.0 ' WILL BE USED IF NO PREVIOUS VALUE WAS ENTERED).
C  ONLY ONE ' 0.' NEEDS TO BE ENTERED IF ALL SCALE FACTORS ARE TO
C  BE LEFT UNCHANGED.
C  NO. OF FILES TO SKIP = -0 TO TERMINATE THE RUN

```

```

      DIMENSION ISKS(12)
      DIMENSION IVSS(12)
      DIMENSION IKEY(6), KEYWD(6)
      DIMENSION DATA(6000), SHIFT(5)
      DATA(IKEYWD(0)=0000454623714465B,0045464751714523B,20202020202020
18,2045466343614447B,2045462263614365B,00000000000634645B)
      COMMON ICODE(12), ISKEW(13), IVC(12), IVS(12), NHI(12),
1 USC(12), VC(12), VCL(12), VSC(12), VSS(12), WSS(12)
      COMMON DEL(5), ID(12), JV(50), POS(5), SAMP(500), X(50)
      DATA(SHIFT = 9, 18, 27, 36, 45), (MAX = 777B), (MIN = -400B)
      COMMON V(6000), RESERVE(12), ID1, ID2, BANG, DANG
      COMMON L90, KEY, NPTS, NLINES, NSA, NSB, KS, NA, NB, KP, NEWR,
1 TITLE(12), TAG1(12), TAG2(12), NSS, NCHAN, KR, CONV, IPOS,
2 IPACK, INT
      EQUIVALENCE (DATA(1), V(1)), (MINSKEW, ISKEW(0))
      INTEGER DATA, DEL, POS, SAMP, SHIFT, TAG1, TAG2, TITLE, VC, X
      LOC(RLB = 13, TEST = 70, ZRO = 0)
      PARTMAP

```

```

CALL CORECON
N5=5
CALL ZERO(VSC, VSC(12), 1.)
ICRASH = 1
FF = 59.0
IEND = 0000000000654564B
LDA(IARG1) SUB(IEND) AJP0(44) . JUMP TO END ROUTINE

LIU1(TEST) IJP1(1) SLJ(2)
1 WRITE(19,101)
101 FORMAT(/ * EXECUTION BEGINS * /)
SLJ(L+2) . SKIP NEXT INSTRUCTION

2 EXF(1000B) SLJ3(3) .IF NOT BATCH, START REAL-TIME CLOCK
+ENA(77776B) STA(IKEY)
STA(IKEY+1) STA(IKEY+5)
ENA(00000) STA(IKEY+2)
DO 500 I=0,2
+ENI6(00006B) LDA1(IARG1)
EQS6(KEYWD) SLJ(503A)
ENA(00001) STA6(IKEY)
SLJ(500) ENI0(0)
503A AJP0(500) STA(IKEY+2)
500 CONTINUE
ENA(00000) STA(IARG1)
STA(IARG2) STA(IARG3) . ZERO OUT PAR FIELDS
+LDA(IKEY+5) AJP3(3)
ENA(00000) STA(IKEY+2)
3 CALL PROCESS(DATA, 1)
LDA(KEY) SUB(5) AJP(91)
CALL ZERO(WSS, WSS(12))
ENA(77776B) STA(IKEY+3)
STA(IKEY+4) STA(IKEY+6)
IFIRST=1
WRITE(9,107) 37B
107 FORMAT(* NEW TITLE* R1/)
LDA(NCHAN) INA(77776B)
+SSK(IKEY) INA(00001)
STA(NC)
IF(NC .LT. 13)11,97
11 CONTINUE
READ(9,108) ID
108 FORMAT(12A8)
LIU1(TEST) IJP1(21) SLJ(22)
21 WRITE(9,108) ID
22 IF(ID(1) .EQ. 3HYES) 11, 12
12 IF(ID(1) .NE. 2HNO) 13, 17

13 DO 14 I=1,12
14 TITLE(I) = ID(I)

17 INT = 1
IUNIT = 3
IPOS2 = 1
ISTART = 0
LSTART = 0
NOTITLE = 0
KBAD = 0
NBAD = 777B
CALL ZERC(NHI, NLO(12))
WRITE(9,117) NC
117 FORMAT(* SKEWS(*I2*I5)= *)
READ(9,103) ISKEW

```

```

103  FORMAT(15I5)
      LIU1(TEST)      IJPI(17A)  SLJ(17C)
17A  WRITE(9,103) (ISKEW(I), I=1,NC)
17C  MINSKEW = 10
      MAXSKEW = -10

      DO 6 J=1,NC
        MINSKEW = MIN0(MINSKEW, ISKEW(J))
        6      MAXSKEW = MAX0(MAXSKEW, ISKEW(J))
        MAXSKEW = MAXSKEW - MINSKEW
        WRITE(9,105)
105  FORMAT(*CLAMP REGION, VCLAMP. 214,12F6.3*/)
      READ(9,106)  NACL, NBCL, VCL
106  FORMAT(214,12F6.4)
      WRITE(9,109)
109  FORMAT(*SCALE REGION, VSCALE. 214,12F6.4 *)
      READ(9,106)  NASS, NBSS, USC
      NBCL=MIN0(NBCL, NSS-MAXSKEW)
      NBSS = MIN0(NBSS, NSS-MAXSKEW)
      IF(NASS .GT.7000) 510,511
510  WRITE(9,520)
520  FORMAT(*SMOOTHING CONSTANT = *)
      READ(9,225) FF
225  FORMAT(F10.0)
      NASS=NASS-7000
511  LDA(NBSS)  ENQ(00001B)
      AJP1(512)  STQ(IKEY+4)
      SLJ(514).
512  SUB(NASS)  AJP3(97)
514  LDA(NBCL)  AJP1(513)
      IPOS2 = IPOS
      AJP1(517)
      KBAD = -255
      NBAD = 255
517  STQ(IKEY+3)  SLJ(24)
513  SUB(NACL)  AJP3(97)
24  NCST = (NACL - 1)*NCHAN
      CONV2 = 0.0
      FVS = 1.0/CCNV
      FFCCNV = FF * CONV
      DCONV2 = 1.0 / ((1.0 + FF) * CONV)
      IKEY(6) = IKEY(4) + IKEY(5) - 1
      N = NC + 1
      LDA(0)  ENQ(00001)
      +THS6(ISKEW)  ENQ(77776B)  .      CHECK FOR ANY DESKEWING
      STQ(JSKEW)

C  DETERMINE ACCEPTABLE LIMITS FOR SIGNALS IN 'TIME' CHANNEL
      NEWR = 0
      +SSK(IKEY+4)  SLJ(18)
      SSK(IKEY)  SLJ(18)
      LDA(IKEY+2)  AJPO(18)
      STA(NSA)
      NSB= NSA+120
      IFIRST=1
      KS=2
      NA= NSS/3
      NB= NA + MIN0(50,NA)
      KP=1
      CALL ZERC(WSS, WSS(12))
194  FSS = NB - NA + 1
      NL = NSB - NSA

```

```

NL = NL/KS + 1
NP = 500/NL
  LIU1(TEST)      IJP1(36)      SLJ(37)
36 WRITE(9,103) NSA, NSB, KS, NA, NB, KP
37 ASSIGN 99 TO N90
  CALL SETEOF(N90)
  CALL FSKIP(0, NSA - NEWR, IUNIT)
  NEWR = NSA
  KM = KS - 1
  LDA(ZRC)        SCL(777777777777000R)
  STA(ISRAN)
  CALL RANIF(ISRAN)
  SLJ4(320)        .SET UP UNPACKING
  ISP = 0
  KR = 0

DO 94 NSCAN = NSA, NSB, KS

  DO 93 K = 1, KR
    NEWR = NEWR + 1
    LDA(IUNIT) SLJ4(RLB).      READ ONE BINARY RECORD
93  CONTINUE

    KR = KM
    NEWR = NEWR + 1
    SLJ4(300).      UNPACK CHANNEL NCHAN INTO X

    DO 94 J = 1, NP
      K = IFIX(RANIF(-1) * FSS - .000001) + NA
      ISP = ISP + 1
94    SAMP(ISP) = X(K)
      SLJ(99A)

99    NSB = NEWR
99A  CALL SORT1(SAMP, ISP, -1)
      NMID = ISP/2
      NQTR = ISP/8
      NLCW = NMID - NQTR
      NHIG = NMID + NQTR
      MIN = SAMP(NLCW)
      MAX = SAMP(NHIG)
41    ANGLE = 3*(MAX - MIN + 1)
      ANGLE = ANGLE/20.
      WRITE(9, 115) MIN, MAX, ANGLE
115  FORMAT(* INTEGER RANGE(* I3 * TO * I3 *) COVERS* F4.1 * DEGREES*/)
42  CALL RSTEOF

18  CALL PROCESS(DATA,2)
      NB = MIN0(NB, NSS-MAXSKEW)
      NB1 = MAX0(NB, NBSS)
      NKP = KP*NCHAN - NC + (ISKEW(1) - ISKEW(NC))*NCHAN
      LIU1(TEST)      IJP1(26)      SLJ(27)
26  WRITE(9,103) NSA, NSB, KS, NA, NB, KP
27  NST = (NA - 1 + ISKEW(1) - MINSKEW) * NCHAN
      NPSS = (NBSS - NASS)/KP + 1
      LDA(NTITLE)      AJP1(46)
      FP = KP
      FANG = (NA - 1)
      BANG = BANG + FANG*DANG
      DANG = FP*DANG
      KSS = (NB-NA)/KP + 1
      MSS = KSS*NC

WRITE(4) TITLE, KSS, BANG, DANG, NC, CONV, IPDS2, 1, TAG1, TAG2

```



```

      +SSK(IKEY+1)  SLJ(32)
46  WRITE(N5, 114) TITLE, KSS, BANG, DANG, NC, CONV, IPOS2, NASS,
      1  NBSS, NACL, NBCL, KP
114  FORMAT(1H1 12A8 / 18 * PTS./SCANLINE,*7X*BANG=* F6.3,7X*DANG= *
      1  F6.3/ 110* CHANNELS,*7X*CONV =*F4.1* (QUANTA/VOLT),*7X*IPOS =*
      2  11/8X*STABLE SOURCE LOCATED BETWEEN POINT NUMBERS*15* AND *15
      3  *, CLAMPING REGION BETWEEN* 15 * AND * 15 *, EVERY* 13 /)
32  VC(0) = 0
      WRITE(9,108) 8HOPTIONS ,8HUSED
      DO 531 J=0,5
      LDA2(IKEY)  AJP3(531)
      WRITE(9,108) KEYWD(J)
531  CONTINUE
      LDA(IKEY+2)  AJP1(532)
      WRITE(9,108) 8HNO TIME , 8HSAMPLING
532  DO 4 J=1,NC
      ICODE(J) = J
      ISKEW(NC-J+1) = ISKEW(NC-J+1) - ISKEW(NC-J)
      VC(J) = VCL(J)*CONV + 0.5
      LDA2(USC)  AJP0(4A)
      VSC(J) = USC(J)
      GO TO 401
4A  USC(J) = VSC(J)
401  VSS(J) = USC(J)
4  IVSS(J) = (VSS(J))*FLOAT(2**15)
      ISKEW(1) = 0

      +SSK(IKEY+1)  SLJ(533)
      +SSK(IKEY+3)  SLJ(29)
      WRITE(N5,121) (VCL(J),J=1,NC)
29  SSK(IKEY+4)  SLJ(29Y)
      WRITE(N5,121) (VSC(J), J=1,NC)
29Y SSK(IKEY+5)  SLJ(29A).
      SLJ(533)
29A  WRITE(N5,121) (VSS(J) ,J=1,NC)

121  FORMAT(7X,12F6.3)
533  DO 7 J=1,NC
      ISKEW(J) = ISKEW(J)*NCHAN + 1
      ISKS(J) = ISKEW(J)
      +SSK(ICRASH)  SLJ(530A)
      CALL EXIT
530A  CONTINUE

      +SSK(IKEY+3)  SLJ(541) . JUMP IF NO CLAMP
      NKPS = 0
      NSTS=0
      NBIS = KSS
      NAS=1
      KPS=1
      NASS = (NASS-NA+1)
      NSST = (NASS-1) * NC
      CALL ZERO(ISKS,ISKS(12),1)
      GO TO 542
541  NSTS = NST
      NKPS = NKP
      NBIS = NB1
      NSST = (NASS-1)*NCHAN
      NAS = NA
      KPS = KP
542  CONTINUE

      CALL BUFFON

```

```

DO 50 NSCAN = NSA,NSB,KS
  CALL PROCESS(DATA, 3)
  LDA(KEY)      SUB(4)      AJP(90)

  CALL ZERO(IVC, IVC(12))
  JSCL = -1
  +SSK(IKEY+4)   SLJ(67)
  LDA(IKEY+2)    AJPO(67)
  SSK(IKEY)      SLJ(67)
  N = (NA-1)*NCHAN

DO 60 I=1,KSS
  M = N
  N = M + NCHAN
  IDAT = DATA(M)
  JDAT = DATA(N)
  LDA(IDAT)      AJP(60)
  SUR(JDAT)      AJP(65)
60  CONTINUE
  GO TO 68

65  IF(IDAT .LE. MAX .AND. IDAT .GE. MIN) 67, 68
68  JSCL = 1
  +SSK(IFIRST)   SLJ(50)
67  K = 0
  +SSK(IKEY+3)   SLJ(501)
  N = NCST

DO 10 IP=NACL,NBCL

DO 9 J=1,NC
  N = N + 1
9   IVC(J) = IVC(J) + DATA(N)

  +SSK(IKEY)     SLJ(10)
  N = N + 1
10  CONTINUE

  NVC = (NBCL - NACL + 1)

DO 30 J=1,NC
30  IVC(J) = IVC(J)/NVC - VC(J)

C  CLAMP THE DATA

  +SSK(IFIRST)   SLJ4(2STCL)
  N = NST
DO 35 IS=NA,NB1,KP

DO 34 J=1,NC
  K = K + 1
  N = N + ISKEW(J)
2CL  LDA6(DATA)    SUB2(IVC)  .CLAMP NORMAL POLARITY DATA
  STA3(DATA)
  +SSK(IKEY+6)    SLJ4(7CST). IF CLAMP ONLY, COLLECT STATISTICS
34  CONTINUE

35  N = N + NKP

501 +SSK(IKEY+5)   SLJ(16)

```

```
521  +SSK(IKEY+4)  SLJ(502)
      SSK(JSCL)  SLJ(16)
```

C COMPUTE SCALE FACTORS

```
      CALL BUFWAIT

      JVMAX = 0
      CALL ZERO(JV, JV(50))
      N = NSST

      DO 19 IP = 1, NPSS

      DO 19 J = 1, NC
        N = N + 1
19     JV(IP) = JV(IP) + DATA(N)

      DO 20 IP = 1, NPSS
20     JVMAX = MAXO(JVMAX, JV(IP))

      K = NPSS + 1
      LDA(JVMAX)
      +EQS3(JV)      SLJ(50).  LOCATE PEAK OF SUN SENSOR
      N = (NASS + (K-2)*KP)*NC

      DO 25 J=1,NC
        N = N + 1
        KVS = WSS(J)*CONV2
        WSS(J) = KVS + DATA(N)
        WSS(J) = WSS(J)*FVS
        AJP(25A)      SLJ(25C)
25A     VSS(J) = 1.
        SLJ(25)
25C     VSS(J) = VSC(J)/WSS(J)
25     CONTINUE
        DO 347 I=1,NC
347     IVSS(J) = VSS(J) * FLOAT(2**15)
```

C SCALE THE DATA

```
16     K = 0
      AN = NSTS

      DO 338 IS=NAS,NBIS,KPS

      DO 340 J=1,NC
        NN = NN + ISKS(J)
        N = NN
        K = K + 1
        LCA6(DATA)  MUI2(IVSS).
        ARS(15)  STA3(DATA).  SHIFT RIGHT AND STORE
        SLJ4(7CST).
340     CONTINUE
338     NN = NN + NKPS

505     IFIRST = -1
        CONV2 = FFCCNV
        FVS = DCONV2
        MMSCAN = NSCAN
```

```

      +SSK(IKEY+1)  SLJ(506).  JUMP IF NOPRINT IS SPECIFIED
      SSK(IKEY+3)  SLJ(15)    . PRINT CLAMP VALUES IF CLAMPING
      WRITE(N5, 102) NSCAN, (IVC(J), J=1,NC)
102  FORMAT(1H , 13I6)
      MMSCAN = 0

      15  +SSK(IKEY+4)  SLJ(506). PRINT SCALE FACTORS IF DYNAMIC SCALING
      WRITE(N5,119) MMSCAN, (VSS(J), J=1,NC)
119  FORMAT(1H ,15,1X,12F6.3)

      506  CONTINUE
      CALL BUFPACK(DATA,MSS)
      50  LDA(KEY)      SUB(3)      AJP(90)

      90  WRITE(9, 125) 378
125  FORMAT(* CONTINUE FILE* R1)
      READ(9, 108) IREP
      IF(IREP .EQ. 3HYES)43, 44
      43  NOTITLE = 1
      GO TO 18
      44  NOTITLE = 0
      ENDFILE 4
      +SSK(IKEY+1)  SLJ(3)
      WRITE(N5,110) TITLE
110  FORMAT(1H1, 12A8)

      WRITE(N5,111) (ICODE(J), J=1,NC)
111  FORMAT(1H0 * CHANNEL* 12I6/)

      WRITE(N5,122) 8H .GT.      , NRAD,(NHI(J),J=1,NC)
122  FORMAT(A8,14,12I6)

      WRITE(N5,122) 8H .LT.      , KBAD,(NLO(J),J=1,NC)

      GO TO 3

      91  ENDFILE 4
      REWIND 4
      LIU1(TEST)      IJP1(95)      SLJ(96)
      95  WRITE(19, 133)
133  FORMAT(* EXECUTION TERMINATED* /)
      RETURN
      96  UNLOAD 3
      ENDFILE N5
      ENDFILE N5
      RETURN
      97  WRITE(9,1001)
1001  FORMAT(* ERROR IN PROGRAM INPUTS. END OF JOB*)
      ICRASH = -1
      N5 = 9
      GO TO 46

      7CST  SLJ(*)  ENIO(0)
      +THS(KBAD)      SLJ(39)
      NLO(J) = NLO(J) + 1
      LDA(KBAD)      SLJ(40)
      39  +THS(NBAD)      SLJ(92)

```

```

      SLJ(7CST)
40   STA3(DATA)
      SLJ(7CST)
92   NHI(J) = NHI(J) + 1
      LDA(NBAD)      SLJ(40)

300   SLJ(*)          .UNPACK 'TIME' CHANNEL INTO X
305   CALL READANY(3, 1, DATA(1), DATA(NWD), NW)
      IF(NW) 98, 306, 307
98   LDA(N90)      SAU(L+1)
      SLJ(*)
306   UNLOAD 3
      PAUSE 12345
      GO TO 305
307   I = ISTART
      L = LSTART

      DO 310 K=1,NB
      LDA4(SHIFT)   SAL(310A)
310A  LDQ1(DATA)    LLS(*)
      SCL(7777777777777000B)   STA3(X)
      I = I + DEL(L)
310   L = POS(L)

      GO TO 300

320   SLJ(*)          .SET UP UNPACKING, AS IN GRAYMAP
      ISTART = (NCHAN + 4)/5
      LSTART = NCHAN + 5*(1 - ISTART)

      DO 325 L = 1, 5
      NEXTPOS = NCHAN + L
      DEL(L) = (NEXTPOS - 1)/5
325   POS(L) = NEXTPOS - 5*DEL(L)

      AWD = (NCHAN*NSS + 4)/5
      GO TO 320

2STCL SLJ(*)   LDA(2CLN).
      +SSK(IVC+2)   LDA(2CLR).   CHECK FIR REVERSE POLARITY DATA
      STA(2CL)     SLJ(2STCL)
2CLN  LDA6(DATA)   SUB2(IVC).   CLAMP NORMAL POLARITY DATA
2CLR  LDA2(IVC)   SUB6(DATA).   CLAMP REVERSE POLARITY DATA

502   SSK(IKEY+3)   SLJ(507).   JUMP IF DESKEWING ONLY
      SLJ(505)
507   LDA(JSKEW)   AJP3(97).   CONTINUE IF DESKEWING ONLY
      N = NST
      K=0
      DO 535 IS=NA,NB1,KP

      DO 534 J=1,NC
      K=K+1
      N = N + JSKEW(J)
      LDA6(DATA)   STA3(DATA)
534   CONTINUE
535   N = N + NKP
      GO TO 505
      END

```

```

      PROGRAM AUTOCAL
C  USES PROCESS, NEW FSKIP, COMMENT, UNPACK3
C  INPUT ON 3, OUTPUT ON 4
C  CORRELATION REFERENCE CHANNEL IS A TYPED INPUT
C  ANSWER 'LINE NO. =' WITH  NSA NSB  KS  NA  NB  BY 515
C  ANSWER 'CORRELATION REGION =' WITH  C1  C2  BY 215
C  THE CORRELATION REGION SHOULD HAVE ELBOW ROOM INSIDE  NA  NB
C  TO PERMIT CORRELATION, AND  NA  NB  SHOULD HAVE ELBOW ROOM
C  INSIDE  1  NSS  TO PERMIT SLEW CORRECTION.
C  A PREVIOUS GRAYMAP OF THE CALIBRATION DATA IS VERY DESIRABLE.
C  SENSE SWITCH 2 ON TO PRINT CHANNEL CORRELATION INFORMATION
C  OUTPUT OF THE PROGRAM IS SLEW-CORRECTED AND DARK-LEVEL CORRECTED
C  AVERAGE LINE, DARK LEVEL, AND INTEGER AND FRACTIONAL CHANNEL SKEW
C  VIDEO DATA USED FOR SLEW COMPUTATION IS NOT DARK LEVEL CORRECTED
C  TO TURN OFF SLEW CORRECTION, MAKE C1 .LT. 0
C  IF KS IS -1, EVERY FIFTH ICHAN POINT ON NSA WILL BE TYPED

      DIMENSION DEL(16),NOFF(16), OFFSET(16), Y(400), PROD(101),Z(2000),
1DARK(16)
      COMMON DATA(6000), X(2000)
      COMMON RESERVE(8), NF, NR, MR, L80, ID1, ID2, BANG, DANG, L90
      COMMON KEY, NPTS, NCHAN, NSA, NSB, KS, NA, NB, KP, IS, TITLE(12),
1 TAG1(12), TAG2(12), NSS, NCHAN, IDA, CONV, IPOS, IPACK, INT
      EQUIVALENCE(X,Z), (Y,PROD), (Y(102),NOFF), (Y(115), OFFSET)
      INTEGER X, Y, DATA, DEL, DELO, DEL1, DEL2, DEL3, TEST, FIRST, AVE,
1 AVEMAX, SUM, SUMO, C1, C2, CX, C2X, CD, DATA1, PROD, PRODMAX
      SET INDEX K1 = L, K2 = M, K3 = N
      DATA(NSKEW = 10)

1  WRITE(9,101)
101  FORMAT(*CORRELATION REFERENCE CHANNEL =*)
      READ(9,102) ICHAN
102  FORMAT(8I5)
      WRITE(9,106)
106  FORMAT(*ARE YOU USING CALIBRATION DATA...*)
      READ(9,107) ICAL
107  FORMAT(A8)
      IF(ICAL.EQ.3HYES) 6, 8
6  ICAL = 1
      GO TO 21
8  ICAL = 0

C  READ AND PRINT TITLE AND OTHER INFORMATION
21  CALL PROCESS(DATA, 1, 1)

C  READ AND PRINT LINE AND COLUMN NUMBERS
2  CALL PROCESS(DATA, 2, 1)

      NVAL = NPTS*NCHAN
      IF(NVAL.GT.2000) 4, 5
4  WRITE(9,104)
104  FORMAT(20HNPTS*NCHAN .GT. 2000)
      GO TO 2
5  CALL ZER0(X, X(NVAL))
      NY = NB - NA + 1
      IF(NY.GT.400) 58, 59
58  WRITE(9,158)
158  FORMAT(*NB - NA + 1 .GT. 400*)
      GO TO 2

```

```

59  WRITE(9,103)
103  FORMAT(*CORRELATION REGION =*)
      READ(9,102) IC1, C2
      C1 = IABS(IC1)
      IF(C1.EQ.0) 12, 13
12   C1 = MIN0(NA+3, NB-3)
      C1 = MAX0(C1, NA)
      C2 = NB - 3
13   C2 = MAX0(C2, C1+1)
      C2 = MIN0(C2, NB)
      WRITE(4,167) C1, C2
167  FORMAT(6X *CORRELATION REGION =* I7, I5/)
      CX = (C1-NA)*NCHAN + ICHAN
      NC = C2 - C1
      C2X = (C2-NA)*NCHAN + ICHAN - NCHAN
      MX = MIN0(C1-NA, NSKEW) + 1

      KINC = NCHAN
      KSTART = (NA-1)*NCHAN
      NA2 = MAX0(NA, 3)
      NB2 = MIN0(NB, NSS-2)
      FIRST = 1
      INT = 1
      NRANGE = MIN0(NSKEW, C1-NA) + MIN0(NSKEW, NB-C2) + 1
      WRITE(4,105)
105  FORMAT(*0 LINE      NA      NB*/ )

C  UNPACK A LINE OF INTEGER DATA INTO THE ARRAY DATA
3    CALL PROCESS(DATA, 3)
      IF(KS.LT.0) 400, 71
71   IF(KEY.EQ.4) 35, 75

C  ESTIMATE ADJACENT-POINT DIFFERENCE ON THE HIGH SIDE
75   LDA(FIRST) AJP(11)
      DO 10 J = 1, NCHAN
        K1 = KSTART + J
        K2 = K1 + KINC
        DO 7 I = 1, NPTS-1
          Y(I) = IABS(DATA(K2)-DATA(K1))
          K1 = K1 + KINC
          K2 = K2 + KINC
7       CALL SORT1(Y, NY, -1)
10    DEL(J) = Y(4*NPTS/5) * 2

C  EDIT NOISE SPIKES OUT OF THE DATA
11   DO 20 J = 1, NCHAN
        K1 = (NA2-1)*NCHAN + J
        K0 = K1 - KINC
        K2 = K1 + KINC
        K3 = K2 + KINC
        DELO = DATA(K0) - DATA(K0-KINC)
        DEL1 = DATA(K1) - DATA(K0)
        DEL2 = DATA(K2) - DATA(K1)
        DO 25 I = NA2, NB2, KP
          DEL3 = DATA(K3) - DATA(K2)
          LDA(DEL1) SCM(DEL2) AJP2(24) .IF DEL1, DEL2 SAME SIGN, TO 24
          TEST = MIN0(IABS(DEL1), IABS(DEL2))
          LEVEL = MAX0(IABS(DELO), IABS(DEL3), DEL(J))
          IF(TEST.GT.LEVEL*5) 22, 24

```

```

22      K1 = K2 - KINC
      DATA(K1) = (DATA(K1+NCHAN) + DATA(K1-NCHAN))/2
24      DELO = DEL1
      DEL1 = DEL2
      DEL2 = DEL3
      K2 = K2 + KINC
25      K3 = K3 + KINC
20      CONTINUE

C  CORRECT FOR SLEW OF THE CALIBRATION DATA
      LDA(IC1)  AJP3(62)
      LDA(ICAL) AJP(62)
      LDA(FIRST) AJP1(31)
      IF(NLINES.EQ.2 .OR. NLINES.EQ.6) 60, 65
60      L = CX
      NL = NLINES - 1
      CD 61 K = 1, NC
      M = L + NCHAN
      Y(K) = (X(M)-X(L))/NL
61      L = M

65      CD = KSTART + CX
      SLJ4(200).  SUM  Y(K) * DEL DATA(L)  FOR ICHAN AT CAL REGION
      SUMO = SUM
      CD = CD + NCHAN
      SLJ4(200).
      IF(SUM.GT.SUMO) 26, 27

26      SUMO = SUM
      NA = NA + 1
      NB = NB + 1
      IF(NB.GT.NSS) 56, 28
56      WRITE(9,157) NA, NB, NSS, IS
157      FORMAT(*CALIBRATION HAS SLEWED OFF END. NA =* I4, *, NB =* I4,
1      *, NSS =* I4, *, LINE NO.* I5)
      GO TO 2
28      CD = CD + NCHAN
      SLJ4(200).
      IF(SUM.GT.SUMO) 26, 31

27      CD = CD - NCHAN
33      CD = CD - NCHAN
      SLJ4(200).
      IF(SUM.GT.SUMO) 32, 31
32      SUMO = SUM
      NA = NA - 1
      NB = NB - 1
      IF(NA.LT.1) 56, 33
31      KSTART = (NA-1)*NCHAN
      NA2 = MAX0(NA, 3)
      NB2 = MIN0(NB, NSS-2)
      IF(SENSE SWITCH 2) 69, 70
69      WRITE(9,131) NA,NB
70      WRITE(4,131) IS, NA, NB
131      FORMAT(3I6)

C  UPDATE THE SUMS OF THE DESIGNATED POINTS
62      FIRST = 0
      K = KSTART
      CD 29 L = 1, NVAL
  
```



```

      K = K + 1
29      X(L) = X(L) + DATA(K)
      IF(KEY.EQ.2) 3, 35

C  AUTOMATIC DESKEWING CALCULATION
35      LC = MAX0(CX - ICHAN - NSKEW*NCHAN, 0)
      DO 50 J = 1, NCHAN
      IF(J.EQ.ICHAN) 50, 40
40      PRODMAX = 740000000000000000B
      LCJ = LC + J
      DO 48 M = 1, NRANGE
      PROD(M) = 0
      L = LCJ
      DO 47 K = CX, C2X, NCHAN
      N = L + NCHAN
      PROD(M) = PROD(M) + (X(K+NCHAN)-X(K)) * (X(N)-X(L))
47      L = N
      IF(PROD(M) .GT. PRODMAX) 41, 48
41      PRODMAX = PROD(M)
      M2 = M
48      LCJ = LCJ + NCHAN

      IF(M2.EQ.1 .OR. M2.EQ.NRANGE) 42, 43
42      BIAS = 0.
      GO TO 44
43      M1 = M2 - 1
      M3 = M2 + 1
      BIASN = PROD(M3) - PROD(M1)
      BIASD = 2 * (2*PROD(M2) - PROD(M1) - PROD(M3))
      BIAS = BIASN/BIASD
44      NOFF(J) = M2 - MX
      OFFSET(J) = FLOAT(NOFF(J)) + BIAS
      IF(SENSE SWITCH 2) 30, 50
30      WRITE(4,150) BIASN, BIASD, BIAS, CX, C2X, LC, LCJ, NRANGE, MX,
1  M1, M2, M3, PRODMAX, (PROD(M), M=1,NRANGE)
150     FORMAT(1H0 3F12.4, 9I8/(8I15))
50      CONTINUE

      NOFF(ICHAN) = 0
      OFFSET(ICHAN) = 0.

C  ESTABLISH DARK LEVEL FOR EACH CHANNEL
      FN = NLINES
      FN = 1./(FN*CONV)
      LDA(ICAL) AJP(72)
      DO 55 J = 1, NCHAN
      SLJ4(300).      COMPUTE DARK LEVEL
55      DARK(J) = FLOAT(AVEMAX)/8. * FN
      WRITE(4,155) (DARK(J), J=1,NCHAN)
155     FORMAT(*ODARK * 12F8.3)
      WRITE(4,156)
156     FORMAT(1H )

C  COMPUTE AVERAGE POINTS SUBTRACTED FROM THE DARK LEVEL
      K = 0
      DO 36 I = 1, NPTS
      DO 36 J = 1, NCHAN
      K = K + 1
36      Z(K) = DARK(J) - FLOAT(X(K))*FN
      GO TO 74

```

```

72   DO 73 K = 1, NVAL
73     Z(K) = FLOAT(X(K))*FN

74   K1 = 1
      K2 = NCHAN
      DO 37 I = NA, NB, KP
        WRITE(4,137) I, (Z(K), K=K1,K2)
137   FORMAT(I6, 12F8.3)
      K1 = K1 + NCHAN
37    K2 = K2 + NCHAN
      WRITE(4,144) (NCFF(J), J=1,NCHAN)
144   FORMAT(*OSKEW= I5, 11I8)
      WRITE(4,145) (OFFSET(J), J=1,NCHAN)
145   FORMAT(6X 12F8.3)
      GO TO 2

200   SLJ(*) .SUM CF Y(K) * DEL DATA(L) FOR ICHAN AT CAL REGION
      SUM = 0
      L = CD
      DO 201 K = 1, NC
        M = L + NCHAN
        SUM = SUM + Y(K)*(DATA(M)-DATA(L))
201   L = M
      GO TO 200

300   SLJ(*). DARK LEVEL COMPUTATION
      AVE = 0
      L = J
      DO 51 I = 1, 8
        AVE = AVE + X(L)
51    L = L + NCHAN
      AVEMAX = AVE
      K = J
      DO 54 I = 9, NPTS
        AVE = AVE - X(K) + X(L)
        K = K + NCHAN
        L = L + NCHAN
54    AVEMAX = MAX0(AVEMAX, AVE)
      GO TO 300

C PROGRAM TO TYPE EVERY FIFTH ICHAN VALUE ON LINE NSA
400   L = ICHAN
      LINC = 5*NCHAN
      LINC2 = LINC + LINC
      DO 401 I = 1, NSS, 10
        WRITE(9,402) I, DATA(L), DATA(L+LINC)
402   FORMAT(I3, 2I5)
401   L = L + LINC2
      GO TO 2
      END
  
```

```

PROGRAM EXPMAP
C VERSION 1.0 2/8/72, ZUK
C NEEDS MAPTRAN, 4-ARG BUFPACK, PROCESS AND UNPACK3
C LIBRARY ON 1, CONTROL DATA ON 2, ADTESTED SCANNER DATA ON 3,
C 2-CHANNEL RECOGNITION AND EXPONENT OUTPUT ON 4, HISTOGRAMS ON 5
C PROVIDES FOR MAPPING WITH PREPROCESSED DATA. FOR UNTRANSFORMED
C DATA, NTRAN = 1 AND DATUM(I,J) = CORRECT SUBSET OF DATA(I,J).
C NT = NO. OF SIGNATURES
C ND = NO. OF CHANNELS ON TAPE
C NN = NO. OF CHANNELS IN SIGNATURE DECK
C NV = NO. OF CHANNELS IN SIGNATURE USED
  
```

```

C  SQ FT = NO. OF SQ FT PER RESOLUTION ELEMENT
C  S(I) = INTEGER REPRESENTING SIGNATURE I
C  TAG(1), TAGA(I) = 16-CHARACTER IDENTIFICATION FOR HISTOGRAMS
C  ISUB = SIGNATURE DECK CORRESPONDS TO A SUBSET OF DATA CHANNELS
C  CODE(J) DEFINES THIS SUBSET
C  JSUB = USED SIGNATURE IS A SUBSET OF THE SIGNATURE DECK
C  ICODE(J) DEFINES THIS SUBSET (ORIGINAL CHANNEL NUMBERS)
C  NTRAN = NO. OF TRANSFORMS COMPUTED. UNTRANSFORMED COUNTS AS 1.
C  NTRAN NEGATIVE TO SIGNAL A SPECIAL OPERATION

      DIMENSION AMAP(6000), H(2280), ITABLE(52)
      COMMON A(20,12), B(12,12,20), D(20), DATA(6000), EIG(12), ID(10),
1  IBIN(21,52), IC(12), ICODE(12), IPT(100), NP(21), S(21), S2(21),
2  TAG(21), TAGA(21), Z(12)
      COMMON CODE(12), DATUM(12), NN, NTRAN
      COMMON R(5), BAN2, DAN2, CC, NF, NR, MR, L80, ID1, ID2, BANG, DANG
      COMMON L90, KEY, NPTS, NLINES, NSA, NSB, KS, NA, NB, KP, IS,
1  TITLE(12), TAG1(12), TAG2(12), NSS, NCHAN, KR, CONV, IPDS,
2  IPACK, INT
      EQUIVALENCE (B, H), (DATA, AMAP)
      INTEGER CODE
      INTEGER AMAP, REPLY, S, TAG, TAGA, TAG1, TAG2, TITLE
      INTEGER SEC, SEC60
      DATA (ITABLE(1)= 511, 490, 480, 470, 460, 450, 440, 430, 420, 410,
1400, 390, 380, 370, 360, 350, 340, 330, 320, 310, 300, 290, 280,
2270, 260, 250, 240, 230, 220, 210, 200, 190, 180, 170, 160, 150,
3140, 130, 120, 110, 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 0)
      LIB(BUFWAIT)
      LOC(TEST = 70)
      PARTMAP

20  FORMAT(20I5)
21  FORMAT(10A8)
22  FORMAT(5E15.8)

      SLJ1(504)
      CALL CORECON
      DUMMY = 0.
501  CALL PROCESS(DATA, 1, 0, 3, 2)
      WRITE(5, 1000) TITLE
      CONV10 = CONV/10.
503  CALL PROCESS(DATA, 2)
      ENA(BUFWAIT)  SAL(L80)
      CALL BUFFON
      IF (MR.GE.0) 17, 642

17  READ(2, 200) NT, ND, NN, NV, LTRAN, SQ FT
200  FORMAT(5I3,3X,5E12.0)
      EXPLIM = 99.6

622  FKS = KS
      FKP = KP
      ACRE = SQ FT * FKS * FKP / 43560.
      NT1 = NT + 1
      ISUB = ND - NN
      JSUB = NN - NV
      NTRAN = IABS(LTRAN)

      IF(ISUB) 305, 306, 305
C  * READ A SUBSET OF DATA CHANNELS IF DESIRED.
305  READ(2,20) ICODE(J), J=1,NN)
      CALL SORT1(CODE, NN, -1)
      GO TO 308
306  DO 307 J = 1,NN

```

```

307     CODE(J) = J

308     IF(JSUB.GT.0) 330, 331
C * READ A SUBSET OF SIGNATURE CHANNELS.
330     READ(2, 20) (ICODE(J), J=1,NV)
        CALL SORT1(ICODE, NV, -1)
        GO TO 337
331     DO 332 J = 1, NV
332     ICODE(J) = CODE(J)

337     L = 1
        DO 340 J = 1, NN
            IF(ICODE(L).EQ.CODE(J)) 341, 340
341     I(L) = J
            L = L + 1
340     CONTINUE

314     S(NT+1)=511
        TAG(NT+1) = 8HNOT CLAS
        TAGA(NT+1) = 6HSIFED

7     LSTART = 1

C * * * * READ SIGNATURE DECKS AND ASSOCIATED INFORMATION * * * *
        DO 108 I=1,NT
            READ(2,21) ID
            IF(ID(1).EQ.4HSAME) 108, 55
C READ SIGNATURE DECK
55     READ(2,22) ((A(I,J),J=1,NN),((B(J,K,I),K=J,NN),J=1,NN))
C READ INTEGER VALUE ASSOCIATED WITH SIGNATURE, NAME OF SIGNATURE,
        READ(2, 600) S(I), TAG(I), TAGA(I)
        600     FORMAT(15,2A8,3F12.0)

C REARRANGE THE MATRIX ACCORDING TO THE SUBSET.

104     DO 2 J=1,NV
        JJ = ICODE(J)

        DO 2 K=J,NV
            KK = ICODE(K)
            B(J,K,I) = B(JJ,KK,I)
2     B(K,J,I) = B(J,K,I)

C CHECK TO SEE IF THE MATRIX IS POSITIVE DEFINITE.
        IF(DIAG(B(0,1,I), EIG, DUMMY, -NV, 12)) 23, 334, 23
23     WRITE(9,124) TAG(I), TAGA(I), (EIG(J), J=1,NN)
124     FORMAT(*MATRIX * 2A8, * IS NOT POSITIVE DEFINITE. EIGENVALUES *
1     *ARE*/ (12E10.3))
        STOP
334     CALL INVERSE(B(0,1,I), B(0,1,I), NV, 12, 12, D(I), ISERR)
        WRITE(5, 123) TAG(I), TAGA(I), S(I), D(I)
123     FORMAT(2A8, 15, E15.5)
336     D(I) = ALOG(D(I))
C
C ARRANGE THE MATRIX B SO THAT IT FITS CORRECTLY INTO THE LINEAR
C MATRIX H.
        L = LSTART + 1
        H(LSTART) = B(1,1,I)
        DO 105 J=2,NV
            J1 = J - 1
            DO 107 K= 1,J1
                H(L) = B(K,J,I) * 2.
107     L = L + 1
        H(L) = B(J,J,I)
  
```

```

105  L = L + 1
108  LSTART = LSTART + 144
      CALL ZERC(NP(1),NP(NT+1))
C
C ** OUTPUT DATA IS ASSUMED TO BE PACKED AND POSITIVE.
      NCHA1 = 2
      NSS1=(NB-NA)/KP+1
      TAG1(1) = 8HRECOGNIT
      TAG2(1) = 8HICN MAP
      TAG1(2) = 8HEXONENT
      TAG2(2) = 8H / 10
      WRITE(4) TITLE, NSS1, BAN2, DAN2, NCHA1, CONV, 1, 1, TAG1, TAG2
652  CALL ZERC(1BIN,1BIN(1092))
      CALL TIMER(3HSET)
C READ AND UNPACK THE INPUT DATA.
642  CALL PROCESS(DATA, 3)
      LDA(KEY)      INA(-4)      AJP(51A)
      CALL TIMER(2HGO)

      I=1
      N=0
      WORD=0

      DO 10 IP = NA, NB, KP
        IPP=(IP-1)*ND
C MAPTRAN TAKES THE DATA (AND IF NECESSARY TRANSFORMS IT) POINT BY
C POINT AND PUTS IT INTO THE SMALL ARRAY DATUM.
        CALL MAPTRAN(DATA(IPP))
        N=N+1
        LSTART = 1
        GMIN = 1.E10
        IMIN = NT1
C * * RECOGNITION CALCULATIONS * *
        DO 12 IM=1,NT
          L = LSTART
          JJ = IC(1)
          Z(1) = DATUM(JJ) - A(IM,JJ)
          SUM = Z(1) * Z(1) * H(L)
          L = L + 1
C
          DO 11 J=2,NV
            JJ = IC(J)
            Z(J) = DATUM(JJ) - A(IM,JJ)
            SUM2 = 0.
            DO 13 K=1,J
              SUM2 = SUM2 + Z(K)*H(L)
13          L = L + 1
11          SUM = SUM + Z(J)*SUM2
          IF(SUM.GT.EXPLI#) 12, 24
24          GM = SUM + D(IM)
          IF(GM.LT.GMIN) 14, 12
14          IMIN = IM
          GMIN = GM
          GSUM = SUM
12          LSTART = LSTART + 144

```

```

C  AMAP IS THE OUTPUT ARRAY
      AMAP(N)=S(IMIN)
      N=N+1
      AMAP(N) = GSUM*CONV10 + .5

C  SEARCH FOR MAKING A HISTOGRAM.
      EN12(52)
      +THS2(1TABLE)  SLJ(900)
900   J=J+1
      IBIN(IMIN,J) = IBIN(IMIN,J) + 1
C  NP = NUMBER OF POINTS RECOGNIZED FOR EACH SIGNATURE.
      NP(IMIN) = NP(IMIN) + 1
10    CONTINUE

C  WRITE THE DATA.
      CALL BUFPACK(AMAP, N, AMAP, 4)
      CALL TIMER(4HSTCP)

C  LINES NOT FINISHED GO TO 642
C  LINES FINISHED GO TO 51
      IF(KEY .EQ. 2) 642, 51

51A   NSB = IS - 1
51    WRITE(9, 1022)
1022  FORMAT(* CONTINUE FILE...*)
      READ(2,1021) REPLY
1021  FORMAT(A1,12A8)
      IF(REPLY .EQ. 1HY) 503, 504
C  ** YES OR Y GO TO 503
504   ENDFILE 4
      CALL BUFFOFF

      CALL TIMER(SEC, SEC60)
      IHRS = SEC/3600
      MINTS = (SEC - 3600*IHRS)/60
      ISEC = (SEC - 3600*IHRS - 60*MINTS)
      NLines = (NSB - NSA)/KS + 1
      WRITE(5, 780) NLines, NSS, ND, NPTS, NV, NT, IHRS, MINTS, ISEC
780   FORMAT(/ * INPUT... * I10* LINES,* I5* PTS/LINE,* I5* CHANNELS*/
1 * PROCESSED... * I5* POINTS,* I5* CHANNELS,* I5* SIGNATURES*/
2 10X *TOTAL TIME= * I5* HRS,* I5* MINUTES,* I5* SECONDS*//)
80    WRITE(5, 32)
32    FORMAT(/ /7X,4HNAME,14X,*VOLTAGE  NUMBER  ACREAGE*//)

      DO 72 J=1,NT+1
      FNP=NP(J)
      AREA=FNP*ACRE
      FS=S(J)
      S2(J)=FS/CONV
      WRITE(5, 81) TAG(J), TAGA(J), S2(J), NP(J), AREA
81    FORMAT( 5X  2A8,3X,F4.1,I12,F12.2)
72    CONTINUE

C  * * * HISTOGRAM * * *

651   DO 660 I=1,NT
      FNP=NP(I)
      AREA=FNP*ACRE

C  WRITE HEADER

      WRITE(5, 1000) TITLE
1000  FORMAT(1H1,10X,12A8)

```

```

        WRITE(5, 1001) TAG(I), TAGA(I), S2(I), NP(I), AREA, D(I)
1001  FORMAT(//7X 4HNAME 14X *VOLTAGE  NUMBER  ACREAGE  LN(DET)* /
        1  5X 2A8, 3X F4.1 11Z, 2F12.2)

        FTAB = (FLOAT(ITABLE(1)))/CONV10
        GTAB = (FLOAT(ITABLE(51)))/CONV10

        WRITE(5, 1002) FTAB, IBIN(I,1), GTAB, IBIN(I,52)
1002  FORMAT(1H *NO. OF POINTS .GT.*F5.1* == 18 *, NO. OF POINTS .LT.*
        1  F5.1 * == 18 // * EXPONENT* 2H**)
        IJK=0

C  DETERMINE THE BIN WITH THE LARGEST NUMBER OF POINTS
        GO 662 J=2,51
        662  IJK = MAX0(IJK, IBIN(I,J))

        ENI2(50)
C  CALCULATE PERCENT IN EACH BIN.
664  IJP2(665)  SLJ(660)
        665  PCT = (FLOAT(IBIN(I,J+2))/FLOAT(NP(I)))*100.
        K=(100*IBIN(I,J+2))/IJK
        CALL ZERC(IPT(1),IPT(100),1H )
        IF(K.GT.0) 667, 668
C  FILL ARRAY AND WRITE.
667  CALL ZERC(IPT(1),IPT(K),1H*)
        668  FTAB = (FLOAT(ITABLE(J+2)))/CONV10
        GTAB = (FLOAT(ITABLE(J+1)))/CONV10
        WRITE(5, 1005) FTAB, GTAB, (IPT(K),K=1,100), PCT
1005  FORMAT(1H F4.1 * TO * F5.1, 100A1 F6.2)
        GO TO 664
        660  WRITE(5, 2006)
2006  FORMAT(// 4H *** CHANNEL 2 IS EXPONENT/10.*)

        65  IF(REPLY .EQ. 1HD) 506, 501
C  IF DONE, GO TO 506...OTHERWISE, GO TO 501

506  ENDFILE 4
        ENDFILE 4
        LIU1(TEST)  IJP1(721)
        ENDFILE 5
        ENDFILE 5
        UNLOAD 2
        UNLOAD 3
        UNLOAD 4
        UNLOAD 5

721  WRITE(19,1030)
1030  FORMAT(* EXECUTION TERMINATED*)
        STOP

```